Naval Research Laboratory

Washington, DC 20375-5000

NRL Memorandum Report 6351

EDITSPEC

A FORTRAN 77 Program for Editing and Manipulating Spectral Data from the Varian CARY 2390 UV-VIS-NIR Spectrophotometer

R. A. BINSTEAD

GEO-CENTERS, Inc.
10903 Indian Head Highway,
Fort Washington, MD 20744

J. C. COOPER

Polymeric Materials Branch
Chemistry Division

October 19, 1988

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Memorandum Report 6351	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Research Laboratory 6b. OFFICE SYMBOL (If applicable) Code 6125	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION SDIO and ONR 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Washington, DC	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 61153N 63224C PROJECT NO: RR011-01-44 TASK NO WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification) EDITSPEC - A FORTRAN 77 Program for Editing and Manipulating Spectral Data from the Varian CARY 2390 UV-VIS-NIR Spectrophotometer

12. PERSONAL AUTHOR(S)
Binstead,* R.A. and Cooper, J.C.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM 10/87 TO present	1988 October 19	70

16. SUPPLEMENTARY NOTATION
*GEO-CENTERS, Inc., 10903 Indian Head Highway, Fort Washington, MD 20744

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	UV-VIS-NIR, FORTRAN, Editing, CARY

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

EDITSPEC is a FORTRAN 77 program designed to provide a number of editing features for UV-VIS-NIR spectral data obtained from the CARY 2300 - 2400 series spectrophotometers. The program is written to operate on a Hewlett-Packard 1000 computer system but with very few system dependent features to enable easy conversion for operation on other host systems. EDITSPEC provides facilities for correction of experimental artifacts and incorrect file descriptors and also includes calculation routines for smoothed, first derivative and second derivative spectra. The program produces new disk files in either the Cary format or in simple ASCII X,Y format for export to curve fitting programs.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Jonn C. Cooper	22b. TELEPHONE (Include Area Code) (202) 767-3115 22c. OFFICE SYMBOL Code 6125

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

CONTENTS

... cont'd

LIST OF TABLES

EDITSPEC

A FORTRAN 77 Program For Editing And Manipulating Spectral Data From The Varian CARY 2390 UV-VIS-NIR Spectrophotometer

INTRODUCTION

Spectral data acquisition from the CARY 2300 and 2400 series spectrophotometers has been implemented successfully with a Hewlett-Packard 1000 minicomputer system using the FORTRAN 77 program, CARYSPEC, and its companion plotting program, PLOTSPEC, which are described in separate reports. The ability to acquire high quality UV-VIS-NIR spectra must be complemented with flexible analytical software to make full use of the spectroscopic data. This report describes a fully tested FORTRAN 77 program, EDITSPEC, which fulfills the data editing and manipulation requirements of most users in a simple to use menu driven environment. The program reads and manipulates disk data files in Absorbance or % Transmission modes *vs* Wavelength (nm) and stores the edited data in the same format or in a simple ASCII X,Y format for use by curve fitting programs.

The editing facilities provided in EDITSPEC allow for easy corrections to both inaccurate file descriptors and anomalies in the spectral data. Single data points may be altered to remove a glitch or a segment of a spectrum may be offset to produce exact matching between regions obtained using different photodetectors or lamp sources. Nonlinearity of Near IR data from the PbS detector can be corrected in cases where the data extend below 800 nm in AUTOSELECT mode. The absorbance offset between the Photomultiplier and PbS detectors at 800 nm is used to rescale the Near IR region assuming that the PM tube response is perfect. The Cary 2300 series instruments can benefit most from this rescaling routine. The Cary 2400 series instruments already provide nonlinearity correction for the standard PbS detector though this feature is bypassed for the reflectance accessory which suffers badly from nonlinearity. EDITSPEC also provides quartic polynomial least squares calculation routines to produce Smoothed, First Derivative and Second Derivative spectra.

EDITSPEC has been developed for use with an HP 1000 minicomputer system running the RTE-6/VM operating system and CI shell. The program resides in a single 32K word memory segment and utilizes 98K words of Extended Memory Addressing (EMA) area for the large data arrays. Since EDITSPEC makes use of very few special features of the HP 1000 computer system the program could be modified easily to run on other host systems supporting the FORTRAN 77 language.

SOFTWARE DESCRIPTION

1.0 Purpose Of EDITSPEC:

The data acquisition program CARYSPEC was designed to store only *baseline corrected* UV-VIS-NIR spectra in disk files which should, hopefully, be free of all instrumental artifacts. In practice, some small irregularities can still occur at grating, lamp or detector change over points and EDITSPEC provides the user with both single datum and region offset commands to remove such artifacts *albeit* in an arbitrary fashion. Discretion is left to the user for determining which spectral region is likely to have been incorrectly recorded by the particular instrument. The Cary 2400 and upgraded 2300 series spectrophotometers are less likely to require such manipulation since these instruments have more reproducible mechanical positioning for the photodetectors.

The data files produced by CARYSPEC include a number of file description strings and numeric variables to provide permanent, internal documentation for the spectrum. Occasionally, important information, such as solute concentration, is not known at the time of data collection and EDITSPEC provides an easy means of displaying and updating the file descriptors. Since EDITSPEC also provides a number of data transformation routines, the edited spectrum can be stored on disk only with a new filename. This prevents inadvertent loss of the original data.

EDITSPEC supports such file operations as file size reduction, nonlinearity correction for Near IR data and numerical procedures for smoothing or experimental noise and calculation of derivative spectra. The file size reduction operation allows new wavelength limits and data step size to be set, which is useful for matching spectra to be used in the difference spectrum mode of PLOTSPEC. Near IR data often suffer from nonlinearity of the photoconductive PbS detectors used in the Cary 2300 series spectrophotometers. Corrections for such nonlinearity are dependent on *both* the light flux and the absorbance level. The Cary 2400 and upgraded 2300 series instruments provide appropriate correction in the Near IR region for the standard PbS detector. However, detectors on external modules, such as the Diffuse Reflectance accessory, are not corrected. Reflectance spectra often show discontinuities between the PbS and PM tube detectors at 800 nm and EDITSPEC provides a simple rescaling procedure to produce detector matching in these cases. This is not meant to be a panacea but it does provide a more realistic result than the raw data.

Experimental noise is not often a problem with UV-Visible spectra from the Cary spectrophotometers. However, the Near IR region is more noise prone, particularly with the Diffuse Reflectance accessory, due to the poorer signal to noise ratio of PbS detectors and the low efficiency of the rather small integrating sphere. In such cases it may be desirable to use the quartic polynomial least squares smoothing routines in EDITSPEC. This standard procedure has been specially modified to suit the large data redundancy in most UV-VIS-NIR spectra and can be used to remove both *random* noise and *spurious* artifacts from solvent overtone bands in the Near IR region.

The polynomial least squares procedure is also utilized in EDITSPEC to calculate First and Second Derivative spectra with a higher degree of precision than the analog versions which the Cary spectrophotometers produce on their internal pen recorders. These routines can provide useful information in regions of overlapping absorption bands. The resulting First and Second derivative spectra are arbitrarily scaled by factors of x10 and x100, respectively, to facilitate subsequent plotting using the program PLOTSPEC.

EDITSPEC also supports the creation of simple ASCII X,Y text files for exporting data to general purpose curve fitting software. In this mode the abscissa values are converted from Wavelength (nm) to Energy units (cm^{-1}) on output with approximately even Wavenumber spacing between points. The user may select either Absorbance or Extinction Coefficient (M^{-1} cm^{-1}) units for the ordinate and the program will automatically rescale % Transmission data to the selected ordinate, if required.

Finally, EDITSPEC provides a facility for subtraction of a reference *baseline* spectrum for cases in which the pure solvent (in transmittance mode) or reflectance standard does not provide a suitable baseline for the sample under study. For example, thin films on substrates or matrix isolated materials will usually require correction for the absorption or reflectivity loss of the substrate. The *baseline* data file may cover the same or a larger wavelength range than the file to be corrected but both must have the same data interval size, ordinate scale and reference beam modes. These restrictions are intended to ensure that such *baseline* corrections are carried out under closely matched instrumental conditions.

The combination of facilities offered by EDITSPEC should satisfy the needs of most users for producing high quality spectra free from instrumental and sample handling artifacts. The operation of each facility will be discussed separately in more detail below.

1.1 Smoothing And Derivative Spectra:

The numerical procedures used in EDITSPEC for smoothing experimental noise and for calculating derivative spectra are based on standard polynomial least squares regression. The method relies on fitting a high order polynomial, $(a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + \ldots)$ to small data segments, $(-x,\ldots,+x)$, replacing the centre point $(x = 0)$ with the fitted values of a_0 (smoothing), a_1 (first derivative), $2a_2$ (second derivative) ... *etc*. The data segment is stepped consecutively across the spectrum to complete the procedure. EDITSPEC uses a quartic polynomial routine with choices of 7, 11, 15, 21 and 25 point segments to provide increasing levels of *local* averaging. The procedure is illustrated below for a 7 point segment:

Index:	-3	-2	-1	0	1	2	3

Smoothing:

The data interval $(-3,\ldots,+3)$ is fitted to a quartic polynomial to calculate a new centre point a_0 denoted by ⊛. The complete spectrum is smoothed by stepping the 7 point segment consecutively across the spectrum but the procedure obviously can not smooth the first or last 3 points. The smoothing procedure may be repeated as many times as desired and may be limited to just a small region of the complete spectrum.

Derivatives:

An obvious extension to the polynomial regression technique is the easy calculation of derivative spectra with replacement of the centre point by the fitted values of a_1, $2a_2$, ... $n!a_n$ *etc*. for the first, second and higher derivatives.

The technique can only average noise (in the least squares sense) within the width of the selected interval so that with closely spaced data even the 25 point fit may be ineffective. For example, Near IR spectra often display the effects of significant 1/f noise with an effective noise period of 20 nm or more underlying a higher level of more random noise. A naive implementation of the smoothing procedure will fail to deal with very low frequency noise, removing only the more rapid fluctuations. EDITSPEC allows the user to span an arbitrarily wide wavelength range for each fitted segment by evenly spacing the fitted points with an integral number of data points. This results in vastly superior *global* averaging.

1.2 Numerical Least Squares Procedures:

The quartic polynomial regression equations can be written in matrix form as follows:

$$\begin{bmatrix} N & \sum_1^N x & \sum_1^N x^2 & \sum_1^N x^3 & \sum_1^N x^4 \\ \sum_1^N x & \sum_1^N x^2 & \sum_1^N x^3 & \sum_1^N x^4 & \sum_1^N x^5 \\ \sum_1^N x^2 & \sum_1^N x^3 & \sum_1^N x^4 & \sum_1^N x^5 & \sum_1^N x^6 \\ \sum_1^N x^3 & \sum_1^N x^4 & \sum_1^N x^5 & \sum_1^N x^6 & \sum_1^N x^7 \\ \sum_1^N x^4 & \sum_1^N x^5 & \sum_1^N x^6 & \sum_1^N x^7 & \sum_1^N x^8 \end{bmatrix} \cdot \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix} = \begin{bmatrix} \sum_1^N y \\ \sum_1^N xy \\ \sum_1^N x^2y \\ \sum_1^N x^3y \\ \sum_1^N x^4y \end{bmatrix}$$

or in matrix notation as $\underline{\underline{C}} \cdot \underline{K} = \underline{B}$, where $k_n = n!a_n$, for which the solution vector is given by $\underline{K} = \underline{\underline{C}}^{-1} \cdot \underline{B}$. Evaluation of the the inverse coefficient matrix is simplified using the range of index values (−x,...,−1,0,+1,...,+x) to represent the data interval to be fitted to the polynomial since all the summations for odd powers of x are zero. The resulting solution vector simplifies to a set of known coefficients times $\underline{B}$ which needs to be evaluated for each data segment. EDITSPEC requires the elements of just the first three rows of $\underline{\underline{C}}^{-1}$, comprising 8 non−zero elements of which 7 are unique. A full pivoting Gaussian elimination matrix inversion program was used to solve for the coefficients and these appear as Double Precision DATA for array CC(I,J) in EDITSPEC for the 5 different fitting algorithms provided.

The general solutions for the smoothing and derivative calculations in EDITSPEC are summarized below:

SMOOTHING: $k_0 = \underline{\underline{C}}^{-1}_{(1,1)} \cdot \Sigma y + \underline{\underline{C}}^{-1}_{(1,3)} \cdot \Sigma x^2y + \underline{\underline{C}}^{-1}_{(1,5)} \cdot \Sigma x^4y$

DERIVATIVE 1: $k_1 = \underline{\underline{C}}^{-1}_{(2,2)} \cdot \Sigma xy + \underline{\underline{C}}^{-1}_{(2,4)} \cdot \Sigma x^3y$

DERIVATIVE 2: $k_2 = \underline{\underline{C}}^{-1}_{(3,1)} \cdot \Sigma y + \underline{\underline{C}}^{-1}_{(3,3)} \cdot \Sigma x^2y + \underline{\underline{C}}^{-1}_{(3,5)} \cdot \Sigma x^4y$

IMPLEMENTATION

2.0 Program Structure:

EDITSPEC comprises a large main program unit containing the console menu displays and string data for the instrument settings variables. A number of subroutines perform input validation and string processing commands absent from the FORTRAN 77 language. The main program unit of EDITSPEC comprises 6 distinct segments of code to carry out the the functions of disk file data retrieval, instrument settings display, editing of file descriptors and spectral data, data file storage and file creation for exporting ASCII X,Y data to curve fitting programs *etc.* The code fragments appear under the following assigned labels: MENU, READ, SETTINGS, EDIT, STORE, CURFILE and EXIT.

MAIN PROGRAM

MENU:

This is the first and main control menu of the program, selecting entry to the data retrieval, instrument settings, editing, data file storage and exit routines. The choices are as follows:

'R' .....Read Spectrum

This command causes a branch to label READ and the program performs a logical test for the presence of a valid spectrum in memory before allowing previous data to be overwritten by a new file. If a spectrum is already present in memory the user may elect to overwrite the old data, subtract a baseline reference file or return to the main menu. The data file format is listed below in Table I. Spectra are read into the data arrays X,Y while baseline files are stored in the arrays XB,YB before subtraction takes place.

'I' .....Instrument Settings

This option branches to label SETTINGS and performs logical tests for the presence of spectrum and baseline data files in memory. This routine allows for display of the most important operating conditions of the Cary 2390 spectrophotometer during acquisition of the selected data file to provide an on-line reference when comparing data files.

'E' .....Edit Spectrum

Entry of this command causes a branch to label EDIT. The program then performs a logical test for the presence of a valid spectrum in memory before presenting a sub–menu of editing options. All the data manipulations are performed from these sub–menus.

'S' .....Store Spectrum

This command transfers control to label STORE where logical tests are performed for the presence of a spectrum in memory and whether the data file to be created is an edited or smoothed spectrum. The data files are stored in the Cary format for subsequent use by the plotting program PLOTSPEC.

'D' .....Store Derivative Spectrum

This command also branches to label STORE but in this case the program also checks for the presence of a derivative array in memory before proceeding. First and second derivative spectra are also stored in the Cary format but are scaled arbitrarily x10 and x100, respectively, to simplify subsequent plotting.

'C' .....Store Curve Fitting File

This option allows for creation of simple ASCII X,Y data files from Cary spectra. The program branches to label CURFILE and prompts for entry of the wavelength range in the new data file and the number of data points required (10 - 500). The abscissa values are converted from Wavelength (nm) to Energy units (cm^{-1}) and the program selects data points to be spaced evenly in cm^{-1}. The ordinate data can be Absorbance or Extinction Coefficients only and automatic rescaling is provided for % Transmission data.

'X' .....Exit

The final option causes a branch to the label EXIT which checks whether the spectrum in memory has been saved to disk before allowing the program to terminate.

TABLE I

Data File Format

Line	File Variables [a]	Format Type [b]
1	LABEL(2)	CHARACTER (A72)
2	DATE(2)	CHARACTER (A8)
3	XMIN(2),XMAX(2),XSTEP(2),CONC(2), PATH(2)	REAL (*)
4	ORD(2),ABSC(2),CELL(2),CYCLE(2), SAMPLE(2),WAVE(2),TIMER(2), TEMP(2),DIST(2)	REAL (*)
5	NP,NV,ND(2)	INTEGER (I3,I3,I6)
6-54	PARAM(2,I)	INTEGER (I2)
55	VARIABLE(2,I)	REAL (*)
56-/	Y(I) or YB(I)	REAL (*)
/-eof	X(I) or XB(I)	REAL (*)

a: Disk data files are read into either half of the data arrays

b: (*) indicates free field format

READ:

This section of EDITSPEC opens a disk file previously stored by the data acquisition program CARYSPEC. The data are stored in ASCII code and contain a complete description of the instrument parameter settings as well as the spectral data. The format of the data file is listed above in Table I. The program prompts the user for both the filename and subdirectory. The filename may be up to 16 characters with the extension ".Sxx", where xx are the researcher's initials. The subdirectory defaults to the user's directory if only a RETURN character is entered. If a directory name is given the program builds a complete CI pathname for use in the OPEN statement. If the file cannot be found or the file is already open the program displays an error message and then returns to the main MENU. Otherwise, the file is read and the file descriptors are displayed on the console while the remainder of the data are being transferred. The program then returns to the main MENU.

SETTINGS:

This section of code displays a list of the most important instrument settings of the Cary 2390 spectrophotometer during acquisition of the specified data file. The program includes a large amount of string data for the various settings in the CHARACTER array Pstr. The INTEGER array PARAM is used as an index to this string data while numeric data for the table are obtained from the REAL array VARIABLE. If both a Spectrum and Baseline are present in memory the program will prompt for which settings to display.

EDIT:

This section of code contains two sub-menu displays for performing all the data and file manipulations within EDITSPEC. The first sub-menu allows for alterations to the file descriptors comprising the variables *Concentration (M)*, *Date (MM/DD/YY)*, *Label (72 chars)* and *Pathlength (cm)* using single letter command entries. The remaining choices select entry to the data editing menu or an exit to the main MENU. The data editing menu supports the following functions: *List Data Segment*, *Edit Single Datum*, *Offset Segment*, *Near IR Rescale*, *Smooth Segment* and *Deriv. Spectra*. The operation of these functions is straightforward and each is described in detail below (Sections 2.1-2.2).

STORE:

This section of EDITSPEC provides the data storage routine for edited, smoothed and derivative spectra. On entry to this routine the program checks if a spectrum is present in memory. Otherwise, an error message is displayed before returning to the main MENU. If the *Store Derivative Spectrum* command has been used to enter this routine then a check is also made for the presence of a valid derivative array. If the data file to be stored is a smoothed or a derivative spectrum then the file description label is altered to include the number of smoothing passes (SMxx: ...) or the degree of the derivative calculation (FD: ... or SD: ...). The user is only prompted for initials, filename and subdirectory information before the file is written to a disk file. Standard I/O error checking for a FILE EXISTS or a FILE OPEN error is performed during file creation. The program then returns to the main MENU with the new filename and file status variables for the edited spectrum.

CURFILE:

This section of code provides a simple means of exporting spectral data to curve fitting programs *etc*. The files are created in an ASCII X,Y format with no other information except the end-of-file marker. The abscissa values are converted from Wavelength to linear Energy units (cm^{-1}) and the program selects data so that the spacing between abscissa values is nearly even in cm^{-1} units. The program provides an arbitrary range of 10 - 500 data points for such files which should suit the needs of most ancillary programs. The ordinate mode can be selected as *Absorbance* or *Extinction Coefficient* only and if the original data were collected in % Transmission mode then automatic rescaling is performed during the data storage loop. Standard I/O error checking is performed for a FILE EXISTS or a FILE OPEN error during file creation. The program then returns to the main MENU with an updated status variable for the *Store Curve Fitting File* command.

EXIT:

The final portion of EDITSPEC checks whether a valid spectrum in memory has been stored to disk before allowing the user to terminate the program. The routine does not check the status of Derivative arrays or Curve Fitting files since these can be easily recreated from the stored spectral data file.

2.1 File Editing Menu Commands:

The first editing sub-menu is designed to allow for easy alterations to the file descriptors. The operations are invoked using single letter commands as follows:

'C' .....Concentration, (M)

The Concentration variable is used to scale experimental absorbance values from solution spectra into Molar Extinction Coefficients (absorptivity) units and may need to be altered if the value was unknown at the time of data collection.

'D' .....Date, (MM/DD/YY)

The Date variable is set manually within the data collection program, CARYSPEC, and may need to be altered if it has been entered incorrectly. Only 8 character positions are provided for this entry.

'L' .....Label, (72 chars)

This variable provides descriptive text to document the nature of the spectrum. The maximum text length is 72 characters and the character field is delimited on the console screen by arrow heads under the text entry line.

'P' .....Pathlength, (cm)

This variable is also used in the calculation of Extinction Coefficient values for solution spectra. In other types of spectra this variable can be set to zero.

The data entry routine is arranged in such a way that the current value of a variable is printed to the console screen with the cursor placed over the first character of the old value. Entry of only a RETURN character will keep the old value intact. If any other characters are entered they are treated as the new value. Thus, a complete line must be entered to update the previous data correctly. Cursor control keys should not be used in editing such line entries - the backspace key is the only editing function permitted.

The remaining command entries 'S' and 'X' are used to enter the data editing sub-menu or to exit to the main MENU.

2.2 Data Editing Menu Commands:

The spectral data editing sub-menu provides access to all the array manipulation functions available in EDITSPEC. In common with the other menus, single letter commands are used to select the desired operations given below:

'L' .....List Data Segment

This command provides a means of inspecting the data between any two selected wavelength limits. The data are displayed in pages of 10 points until completion. If the original data were acquired in Absorbance mode this routine lists both the Absorbance and Extinction Coefficient values *vs* Wavelength (nm). In % Transmission mode only the raw data are presented.

'E' .....Edit Single Datum

This command is used to display and update Absorbance or % Transmission data at a single wavelength. The current value is printed on the screen with the cursor positioned over its first character. Entry of only a RETURN character will keep the current value intact. Otherwise, a new entry is assumed for any other characters read from the keyboard. New entries are validated to lie within the range −0.5 to 4.5 for Absorbance mode and 0 to 200 for % Transmission mode. The latter also covers the case of % Reflectance spectra.

'O' .....Offset Segment

The offset command enables corrections to be made for misalignment between regions of a spectrum obtained using different detectors, gratings or source lamps. The routine provides a simple (±) algebraic offset, selected by the user, between the desired wavelength limits without regard to the ordinate mode.

'N' .....Near IR Rescale

This command is used to align the Near IR region of a spectrum with data obtained from the photomultiplier detector in the Visible region. The data must, therefore, have been collected in AUTOSELECT mode so that the detector change is made at 800 nm. The routine inspects the last point obtained with the PbS detector and the first point from PM tube to obtain a default offset value. The default value is written to the console display with the cursor positioned over its first character and entry of just a RETURN character will commence rescaling with this offset.

This routine is most often used to align noisy, high absorbance Near IR data with a noise free Visible region so that it proves more useful to plot the data first to obtain a better estimate for the offset before rescaling. The rescaling procedure is multiplicative for both Absorbance and % Transmission modes. Ideally, the latter should be converted to Absorbance before rescaling but this is not likely to cause any serious errors.

'R' .....Reduce File Size

This command is used to change the size of a data file by reducing the wavelength range or by increasing the step size (nm) between data points. This is most useful in providing matched data files for the spectral subtraction mode of the plotting program, PLOTSPEC. It may also be useful for eliminating excess data from very large files. The reduced file overwrites the current spectrum and recovery is possible only by reading the old data from disk.

'S' .....Smooth Segment

The quartic polynomial smoothing routine in EDITSPEC is provided with data for calculations with 5 different levels of *local* averaging using data segments with 7, 11, 15, 21 or 25 points (-XX,...,0,...,+XX) in the least squares fits. These are selectable from the smoothing algorithm menu which is presented on entry to the routine. The program then displays the wavelength range of the fitted segments with a default integer step size of two points. The user may increase or decrease the wavelength range of the fitted segments by altering the step size variable. Larger values of the step size produce improved *global* averaging, an important feature for removing non-random solvent overtone artifacts. The smoothing routine may be applied over the whole wavelength range or just within a small segment. However, the routine can not smooth the first or last XX data points in the spectrum. The program prints the actual range smoothed on the console screen before prompting for the number of smoothing passes. There is nothing to be gained by performing a large number of smoothing passes with the same algorithm. Rather, it is preferable to select several different fitting intervals, by choice of algorithm and step size, to improve the averaging of low frequency, non-random noise. In general, it is wise to finish a smoothing procedure with the integer step size set to 1 data point to suppress a tendency to produce periodic oscillations in the data with the polynomial regression procedure.

Some examples of smoothed spectra appear as an appendix to this report.

'D' .....Derivative Spectra

The polynomial regression technique in EDITSPEC is also used to calculate first and second derivative spectra. Smoothing may be performed prior to the derivative calculations in order to reduce the noise level, particularly for the second derivative. Selection of the fitting algorithm is identical to the procedure for smoothing. However, the procedure can be performed only once and must encompass the entire wavelength range, less the first and last XX data points. The latter are set equal to the first and last fitted points, respectively. It is most beneficial to provide wide *global* averaging for the derivative functions with fitted intervals on the order of 20-50 nm. If the fitted interval is too narrow then the result is a measure of the point to point noise level and *not* the derivative of the spectrum *per se*.

Some examples of derivative spectra appear as an appendix to this report.

2.3 COMMON DATA:

All COMMON array variables used by EDITSPEC are held in a named COMMON block, /DATA/. The arrays contained within the COMMON block are listed below:

/DATA/ Contains EMA REAL Arrays of spectroscopic data

X
Array containing the spectrum's Wavelength data (nm)
Values read from disk data file
Values used in main program unit

Y
Array containing the spectrum's Absorbance or %T data
Values read from disk data file
Values used in main program unit

XB
Array containing the baseline's Wavelength data (nm)
Values read from disk data file
Values used in main program unit

YB
Array containing the baseline's Absorbance or %T data
Values read from disk data file
Values used in main program unit

Z
Array containing the derivative spectrum's Absorbance or %T data
Values calculated in main program unit
Values stored to disk data file

All the data arrays are dimensioned for 10001 data points maximum in single precision and are specified to reside in EMA memory requiring 98K words of system available memory. These memory demands could be reduced if required since few UV-VIS-NIR spectra have more than a few thousand data points.

2.4 SUBROUTINES:

EDITSPEC uses subroutines to perform specific tasks which are required more than once, including several types of string manipulation and input validation. The purpose and calling sequences are listed below:

Center(TITLE)

Prints a string on the user console centred within a 72 column line.

TITLE CHARACTER*72 string, contents set by calling unit

CALLED BY: Main program unit
CALLS: None

Line(N)

Prints a line of '–' characters to the user console N columns wide and centred within a 72 column line.

N INTEGER variable input from calling unit

CALLED BY: Main program unit
CALLS: None

Xlimits(MIN,MAX)

Performs an absolute value function on the input arguments
Validates the resulting arguments so that MIN < MAX

MIN,MAX REAL variables input and output with order swapped if required

CALLED BY: Main program unit
CALLS: None

Ylimits(MIN,MAX)

Validates the input arguments so that MIN < MAX

MIN,MAX REAL variables input and output with order swapped if required

CALLED BY: Main program unit
CALLS: None

Upper(Code)

Performs a check for lower case characters in a string of arbitrary length and converts to upper case if necessary.

Code CHARACTER*(*) variable passed into routine and UPPER case on exit DIMENSION is set by the calling unit

CALLED BY: Main program unit
CALLS: None
EXTENSION: LEN(*string*) function, an HP extension to FORTRAN 77

Str(VALUE,String,PREC)

Performs a conversion from numeric value to a string number for floating point numbers only with up to 12 digits precision.

VALUE REAL variable input to be processed by the routine

String CHARACTER*14 string output corresponding to VALUE

PREC INTEGER variable input to set the rounding precision for string

CALLED BY: Main program unit
CALLS: None

Val(String,VALUE)

Performs a conversion from string to numeric value for a string number containing up to 10 digits with or without an exponent.

String — CHARACTER*(*) string input to be processed by routine
DIMENSION is set by calling unit

VALUE — REAL variable output

CALLED BY: Main program unit
CALLS: None

Wait(DELAY)

Performs a loop which tests the system clock until DELAY seconds have elapsed. The routine does not make provision for the special case at the transition to 2400 hours.

DELAY — REAL variable holding the value of the delay period in seconds

CALLED BY: Main program unit
CALLS: FUNCTION Time(I)

2.5 FUNCTIONS:

EDITSPEC uses only one function subprogram that makes an EXEC call to read the system time.

Time(I)

Performs an EXEC call to read the system clock and converts the reading to seconds and centiseconds.

I Dummy argument

CALLED BY: SUBROUTINE Wait only
CALLS: EXEC(ICODE,ITIME) system level command

2.6 SPECIAL SYSTEM CALL:

EDITSPEC makes use of a system subroutine call to alter the width of data fields sent to the system console. The record length is changed from the default value of 72 columns to a more useful value of 79 in order to prevent inadvertent screen wrap around.

FFRCL(I)

Alters the column width for standard output

I Default value = 72, EDITSPEC value = 79

CALLED BY: Main program unit only
CALLS: None, system level subroutine call, syntax CALL FFRCL(I)

PROGRAM CODE

3.0 Source Code Availability:

The source code for the program EDITSPEC is a 65K ASCII text file available on either a Hewlett-Packard cartridge, 9 track tape or an IBM 360K format floppy disk. All requests should be accompanied by the blank medium desired. A printed copy of the source code is listed below.

3.1 Variable Names And Usage:

A complete listing of the INTEGER, REAL, REAL Array and CHARACTER variables for the MAIN segment of EDITSPEC is given below in Tables II, III, IV & V, respectively. The subroutines use the same names as the main program for the same variables. Additional variables in the subroutines and simple integers, I-N, are not documented since their usage is rather obvious. The logical variable MATCH is used within the program when comparing two spectra for the baseline reference subtraction mode. The logical variable DERIV controls the operation of the smoothing routine in order to re-use the algorithm selection menu and then branch back to the derivative calculation routine.

Table II

Glossary of INTEGER Variables

Name	Description	Value
CURFILE	Assigned Label - Curve Fitting File	2000
EDIT	Assigned Label - Editing Sub-menus	400
EXIT	Assigned Label - Terminate Program	9999
MENU	Assigned Label - Main Control Menu	10
READ	Assigned Label - Read Spectrum, Baseline	100
SETTINGS	Assigned Label - Scan Conditions	300
STORE	Assigned Label - Store Spectrum	1000
ASCII	ASCII equivalent of digits in Str	48-57
COUNT	Number of smoothing cycles performed	>=0
FINISH	Index of final abscissa value to edit	1-10001
IS	Step size in smoothing or derivatives	1-ITEST
ITEST	Maximum step size for selected algorithm	see text
NCOL	Number of screen columns in menu display	50-70
NDATA	Number of data points in spectrum	1-10001
NFIT	Number of points in curve fitting file	10-500
NPASS	Number of smoothing passes to execute	>=1
NP	Number of parameters to read from file	49
NV	Number of variables to read from file	14
START	Index of first abscissa value to edit	1-10001
XOFF	Index offset at start & end of spectrum	see text
XX	Linear convolution values in least squares	see text
ND(2)	Number of data points read from disk file	1-10001
NPTS(5)	Number of points in least squares fit	7-25
PARAM(2,49)	Instrument operating modes table	1-16

Table III

Glossary of REAL Variables

Name	Description
BAND	Spectral Bandwidth (nm) - AUTO GAIN mode
EC	Extinction Coefficient
FIRST	Starting wavenumber value in X,Y file
GAIN	Instrument gain - AUTO SLIT mode
LAST	Ending wavenumber value in X,Y file
NUMBER	General purpose data entry variable
PMIN	Pen scale minimum limit
PMAX	Pen scale maximum limit
RANGE	Wavelength span for least squares fit
SPACE	Approximate wavenumber steps in X,Y file
STEP	Step size (nm) interval, editing mode
W	Wavenumber loop counter in X,Y file creation
WL	Wavelength to be edited or displayed *etc.*
WMIN,WMAX	Wavelength limits for spectrum in memory
WN,WX	Current wavelength limits for editing
WOFF	Wavelength offsets for smoothing & derivatives
Y1,Y2	Ordinate values at Near IR, Visible change over
YOFF	Absorbance or %T offset, editing mode
ZM	Multiplier for 1st & 2nd derivatives

Table IV

Glossary of REAL Array Variables

Name	Description
ORD(2)	Final ordinate value in file
ABSC(2)	Final abscissa value in file
CELL(2)	Final cell # value in file
CYCLE(2)	Final cycle # value in file
SAMPLE(2)	Final sample # value in file
WAVE(2)	Final wavelength value in file
TIMER(2)	Final time value in file
DIST(2)	Final distance value in file
CONC(2)	Concentration of sample (M), from data file
PATH(2)	Pathlength of sample cell (cm), from data file
VARIABLE(2,14)	Instrument operating conditions table
XMAX(2)	Starting wavelength of scan (nm)
XMIN(2)	Ending wavelength of scan (nm)
XSTEP(2)	Step size (nm), from data file
X(10001)	Wavelength array - spectrum
XB(10001)	Wavelength array - baseline
Y(10001)	Absorbance, %T array - spectrum
YB(10001)	Absorbance, %T array - baseline
Z(10001)	δ(Abs), δ(%T) array - derivative spectrum

Table V

Glossary Of CHARACTER Variables

Name	Description
Screen Control:	
BELL	CHAR(7) bell character
CLR*2	Clear screen
DOWN*2	Move cursor down 1 line
ESC	CHAR(27) escape character
HOME*2	Move cursor to upper left corner
UP*2	Move cursor up 1 line
File Status:	
Cstat*10	Curve fitting file status (STORED)
Dstat*10	Derivative spectrum status (STORED)
DATE(2)*8	Date (mm/dd/yy)
Fstat*10	Edited spectrum status (VALID, STORED)
LABEL(2)*72	Descriptor of spectrum
Pstr(49,16)*14	Table of parameter setting names
Einc*4	Step size interval (nm) - edited spectrum
Emax*4	Starting wavelength (nm) - edited spectrum
Emin*4	Ending wavelength (nm) - edited spectrum
Sinc*4	Step size interval (nm)
Smax*4	Spectrum starting wavelength (nm)
Smin*4	Spectrum ending wavelength (nm)
Sname(2)*20	Spectrum pathname, CI convention
Sstat(2)*10	Spectrum status (VALID, INVALID)

/...

...cont'd

Program Control:

C	Literal comma ','
Code	Menu selection, valid until reset
Directory*40	User directory name, CI convention
Dtype	Type of derivative spectrum (F or S)
Filespec*63	Full pathname for file retrieval or storage
Fit(5)*2	Number of points used by fitting algorithm
Fname*20	Filename entry for building Filespec
Icode	General purpose selection key entry
HEADER*72	Combined file label for data storage
Pfit*10	Literal 'POINTS FIT'
String*14	String to pass data to or from subroutines
TITLE*72	String to be printed to screen
Ymode	Disk file ordinate mode - (A or T)

```
1 FTN7X,L
2 $FILES 0,1
3 $EMA/DATA/
4       PROGRAM EDITSPEC
5 C     ------------------------------------------------------------------
6 C
7 C     This Program Is Designed To Edit Spectral Data Acquired From
8 C     The CARY 2390 UV-VIS-NIR Spectrophotometer With CARYSPEC.RUN
9 C
10 C    The Program Provides Essential Features For Removing Artifacts
11 C    From Experimental Data With Commands For Editing Single Points,
12 C    Offsetting Large Wavelength Regions And Rescaling Near IR Data
13 C    To Correct Mismatch Between The PbS And PM Tube Detectors. The
14 C    Program Also Provides Least Squares Routines For Calculation Of
15 C    Smoothed, First Derivative And Second Derivative Spectra.
16 C
17 C    ------------------------------------------------------------------
18 C
19 C        AUTHOR:  Dr. Robert A. Binstead,
20 C                 Chemistry Division, Code 6125,
21 C                 Naval Research Laboratory,
22 C                 Washington. D.C. 20375
23 C                 U.S.A.
24 C
25 C        WRITTEN: March, 1987
26 C
27 C        VERSION: 1.8
28 C
29 C        REVISED: April, 1987  - Improved Smoothing Routine To Allow
30 C                                For Multiple Data Steps Between The
31 C                                Abscissa Points In Each Fit.
32 C                              - Added First And Second Derivative
33 C                                Spectrum Routines Based On The Same
34 C                                Quadratic Polynomial Routine.
35 C
36 C                 May,    1987 - Added File Size Reduction Routine
37 C                                To Enable Exact Matching Of Spectra
38 C                                For Difference Plotting Mode In The
39 C                                Program PLOTSPEC
40 C
41 C                 June,   1987 - Modified File Name Convention To
42 C                                Match The Use Of Directories By
43 C                                The CI Operating System
44 C
45 C                 August,1987  - Added Data File Creation Routine
46 C                                For BANDPLOT Curve Fitting Of
47 C                                Multiple Gaussians To Spectrum
48 C
49 C                 October,1987 - Added Baseline Correction Routine
50 C                                For Point By Point Subtraction Of
51 C                                A Second File
52 C                              - Added A NIR Scaling Routine To
53 C                                Match Reflectance Data From The PbS
```

```
C                                   And PM Tube Detectors At 800 nm
C
C                     January,1988 - Added Upper Case Conversion Routine
C                                    To Permit Case Insensitive Input
C
C                     May, 1988    - Revised Data File Creation Routine
C                                    To Use The Simpler (X,Y) Format
C                                    Required By BANDFIT, An Interactive
C                                    Version Of The Old BANDPLOT Program
C                                  - Removed Unused Block Data & Arrays
C                                    Removed Unused Variables & Trimmed
C                                    Down String Data Arrays
C                                  - Added Facility To Edit Data Files
C                                    Collected In % Transmission Mode
C
C                     June, 1988   - Added 25 Point Algorithm To The
C                                    Smoothing & Derivative Routines
C
C          Y-AXIS:  Absorbance
C
C          X-AXIS:  Wavelength
C
C          MEMORY:  30,000 Words (PROGRAM) + 98,000 Words EMA (DATA)
C
C     ------------------------------------------------------------------
C
      INTEGER COUNT,CURFILE,EDIT,EXIT,FINISH,IS,ITEST,MENU
      INTEGER NCOL,NDATA,NFIT,NPASS,ND(2),NPTS(5),NP,NV,PARAM(2,49)
      INTEGER READ,SETTINGS,START,STORE,XOFF,XX
      REAL*8 CC(5,7),SS(5),YY
      REAL BAND,EC,FIRST,GAIN,LAST,NUMBER,PMIN,PMAX,RANGE
      REAL SPACE,STEP,VARIABLE(2,14),W,WL,WN,WX,WOFF,Y1,Y2,YOFF,ZM
      REAL ABSC(2),CELL(2),CONC(2),CYCLE(2),DIST(2),ORD(2),PATH(2)
      REAL SAMPLE(2),TIMER(2),TEMP(2),WAVE(2),XMIN(2),XMAX(2),XSTEP(2)
      REAL X(10001),Y(10001),XB(10001),YB(10001),Z(10001)
      LOGICAL DERIV,MATCH
C
C         Dimension Screen Control String Variables
C
      CHARACTER BELL,CLR*2,DOWN*2,ERASE*2,ESC,HOME*2,UP*2
C
C         Dimension Program Parameter Variables
C
      CHARACTER Cstat*10,Dstat*10,Fstat*10,Sstat(2)*10
      CHARACTER Directory*40,Filespec*63,Fname*20,Sname(2)*20
      CHARACTER Smin*4,Smax*4,Sinc*4,Emin*4,Emax*4,Einc*4
      CHARACTER C,Code,DATE(2)*8,Dtype,Fit(5)*2,HEADER*72,Icode
      CHARACTER INITIALS*2,LABEL(2)*72,Pcode*2,Pfit*10,Pstr(49,16)*14
      CHARACTER String*14,TITLE*72,Ymode
C
C     ------------------------------------------------------------------
C
      COMMON /DATA/X,Y,XB,YB,Z
```

```
C
C     ---------------------------------------------------------------
C
C         Smoothing & Derivative Routine Constants For Quartic Fit
C
C         Smoothing       : J = 1,3,5
C         1st Derivative  : J = 2,4
C         2nd Derivative  : J = 3,6,7
C
C         I : Segment Size : Abscissa Range : Steps : Data Range
C                                           (MIN=1)
C         1         7          -3,...,+3        X     -3X,...,+3X
C         2        11          -5,...,+5        X     -5X,...,+5X
C         3        15          -7,...,+7        X     -7X,...,+7X
C         4        21         -10,...,+10       X    -10X,...,+10X
C         5        25         -12,...,+12       X    -12X,...,+12X
C
C         Smoothing Of Experimental Spectra Is Almost Universally
C         Performed Using A Polynomial Least Squares Approximation
C         Within Small Data Intervals, Stepped Consecutively Across
C         The Spectrum. The Technique Must Be Applied Intelligently
C         Since No Significant Smoothing Will Occur If The Size Of
C         The Smoothing Segments Is Smaller Than The Period Of The
C         Noise. This Is A Particular Danger With Spectra From The
C         Cary 2390, Where Data Are Often Acquired In Small Steps.
C
C         This Program Utilizes Two Complementary Approaches To
C         Increase The Wavelength Interval Of The Fitted Segments.
C         The First Includes More Data Points Within The Segment With
C         Choices Of 7,11,15,21 & 25 Point Quartic Polynomial Fits To
C         Provide Increasing Levels Of LOCAL Averaging. The Second
C         Method Utilizes Multiple Data Steps Between Each Point In
C         The Fitted Segments, Allowing For Arbitrarily Wide GLOBAL
C         Averaging.
C
C         Random Noise Of Shorter Period Than The Fitted Segments Is
C         Well Filtered By The Smoothing Routine. However, Non-Random
C         Noise Can Produce Sinusoidal Smoothed Spectra. One Means Of
C         Suppressing These Anomalies Is To Apply Several Cycles Of
C         Smoothing, Each With A Different Segment Size Or With A
C         Different Number Of Data Points In The Fits. Small Regions
C         Of Non-Random Artifacts Are Best Removed Using A Large Step
C         Size And Large Number Of Fitted Points For Just That Region
C         Followed By Smoothing The Whole Spectrum With Unit Step Size.
C
C         The Use Of Polynomial Curve Fitting For First & Second
C         Derivative Spectra Is Subject To Digitizing Noise With
C         Data Segments Which Cover Only A Small Wavelength Range.
C
C         The 7,11,15,21 & 25 Point Algorithms Provide Increasing
C         Levels Of Smoothing At The Expense Of Spectral Resolution.
C         This Should Not Be A Problem As Data Are Usually Acquired
C         At Very Small Increments (Large Redundancy). However, The
```

```
C         Larger Range Algorithms Tend To Produce A Sinusoidal Fit,
C         Resulting In Very Strange Second Derivative Spectra. This
C         Can Be Largely Eliminated By Use Of A Minimum Of 2 Data
C         Points Between Fitting Points, While Stepping The Routine
C         1 Data Point At A Time Across The Spectrum.
C
C         A Benefit Of Increased Spacing Between Abscissa Values Is
C         Better Averaging - This Also Reduces Spectral Resolution
C         But Is Still Far Better Than ANALOG Derivative Spectra.
C
C         It Is Recommended That Smoothed, First & Second Derivative
C         Spectra Be Obtained With Increasing Step Size, e.g. 2:4:8,
C         Since Greater Averaging Is Required For The Derivatives.
C
C         A Typical Broad Solution Spectrum Will Require A Fitting
C         Segment Of 25 nm To Be Effective For Second Derivatives.
C         The User Must Decide Which Algorithm & Step Multiple Is
C         Most Appropriate For The Particular Spectrum But A General
C         Guideline Is To Select A Fitting Interval In Which There
C         Is A Significant Change In Absorbance Or Slope.
C
C     ------------------------------------------------------------------
      DATA (NPTS(I),I=1,5)/7,11,15,21,25/
      DATA (CC(1,J),J=1,7)/5.6709956709957D-1,2.6256613756614D-1,
     &-2.6515151515151D-1,-3.2407407407407D-2,2.2727272727273D-2,
     &2.1433080808081D-1,-2.1148989898990D-2/
C     ------------------------------------------------------------------
      DATA (CC(2,J),J=1,7)/3.3333333333333D-1,6.0379435379435D-2,
     &-5.5361305361305D-2,-2.8813778813779D-3,1.7482517482517D-3,
     &1.6341297591298D-2,-6.0703185703186D-4/
C     ------------------------------------------------------------------
      DATA (CC(3,J),J=1,7)/2.395159020546D-1,2.3045899271389D-2,
     &-2.0648855788175D-2,-5.8306798502877D-4,3.4099027907077D-4,
     &3.1838914869101D-3,-6.2206670178806D-5/
C     ------------------------------------------------------------------
      DATA (CC(4,J),J=1,7)/1.6923254427629D-1,8.2485070089952D-3,
     &-7.3024633194239D-3,-1.0562014757286D-4,6.0573428456051D-5,
     &5.6541148461326D-4,-5.5661610043088D-6/
C     ------------------------------------------------------------------
      DATA (CC(5,J),J=1,7)/1.4169580127478D-1,4.8623545000357D-3,
     &-4.2895207765503D-3,-4.3823594548232D-5,2.4987497756926D-5,
     &2.3322878661008D-4,-1.6138972424588D-6/
C
C     ------------------------------------------------------------------
C
C         Initialize String Variables
C
C     ------------------------------------------------------------------
C
      DATA (Pstr(1,I),I=1,6)/'ABSORBANCE','% TRANSMISSION',
     &'TEMPERATURE','% REFLECTANCE','CONCENTRATION','EMISSION'/
      DATA (Pstr(2,I),I=1,4)/'WAVELENGTH','TIME','TEMPERATURE',
     &'DISTANCE'/
```

```
213       DATA (Pstr(3,I),I=1,11)/'OFF','0.01','0.02','0.05','0.1','0.2',
214      &'0.5','1.0','2.0','5.0','10.0'/
215       DATA Pstr(4,1)/'OFF'/
216       DATA (Pstr(4,I),I=6,15)/'0.2','0.5','1.0','2.0','5.0',
217      &'10','20','50','100','200'/
218       DATA (Pstr(5,I),I=1,4)/'AUTO SELECT','AUTO GAIN','AUTO SLIT',
219      &'SINGLE BEAM'/
220       DATA (Pstr(6,I),I=1,5)/'OFF','NORMAL','1ST DERIV','2ND DERIV',
221      &'LOG'/
222       DATA (Pstr(7,I),I=1,9)/'0.01','0.02','0.05','0.1','0.2','0.5',
223      &'1.0','2.0','4.0'/
224       DATA (Pstr(8,I),I=10,16)/'2','5','10','20','50','100','200'/
225       DATA (Pstr(9,I),I=12,15)/'10','20','50','100'/
226       DATA (Pstr(10,I),I=1,5)/'-1.9 TO 0.6','-2.0 TO 0.5',
227      &'-2.1 TO 0.4','-2.2 TO 0.3','-2.3 TO 0.2'/
228       DATA (Pstr(11,I),I=1,16)/'+/-0.01','+/-0.02','+/-0.05','+/-0.1',
229      &'+/-0.2','+/-0.5','+/-1.0','+/-2.0','+/-5.0','+/-10','+/-20',
230      &'+/-50','+/-100','+/-200','+/-500','+/-1000'/
231       DATA (Pstr(12,I),I=1,16)/'+/-0.01','+/-0.02','+/-0.05','+/-0.1',
232      &'+/-0.2','+/-0.5','+/-1.0','+/-2.0','+/-5.0','+/-10','+/-20',
233      &'+/-50','+/-100','+/-200','+/-500','+/-1000'/
234       DATA (Pstr(13,I),I=1,8)/'+/-0.01','+/-0.02','+/-0.05','+/-0.1',
235      &'+/-0.2','+/-0.5','+/-1.0','+/-2.0'/
236       DATA (Pstr(14,I),I=1,10)/'0','10','20','30','40','50','60','70',
237      &'80','90'/
238       DATA (Pstr(15,I),I=1,4)/'0.5','1.0','3.0','10'/
239       DATA (Pstr(16,I),I=1,2)/'NORMAL','REVERSE'/
240       DATA (Pstr(17,I),I=1,2)/'OFF','ON'/
241       DATA (Pstr(18,I),I=1,2)/'REPEAT SCAN','SGL/MULTI'/
242       DATA (Pstr(19,I),I=1,2)/'SERIAL','OVERLAY'/
243       DATA (Pstr(20,I),I=1,4)/'BOTH ON','UV ONLY','VIS/NIR ONLY',
244      &'BOTH OFF'/
245       DATA (Pstr(21,I),I=1,3)/'AUTO','UV','VIS/NIR'/
246       DATA (Pstr(22,I),I=1,3)/'AUTO','UV','VIS/NIR'/
247       DATA (Pstr(23,I),I=1,2)/'FULL','1/3'/
248       DATA (Pstr(24,I),I=1,3)/'AUTO','UV/VIS','NIR'/
249       DATA (Pstr(25,I),I=1,3)/'AUTO','UV','VIS/NIR'/
250       DATA (Pstr(26,I),I=1,4)/'AUTO SELECT','AUTO GAIN','AUTO SLIT',
251      &'SINGLE BEAM'/
252       DATA (Pstr(27,I),I=1,2)/'FULL','1/3'/
253       DATA (Pstr(28,I),I=1,6)/'0','1','2','3','4','5'/
254       DATA (Pstr(29,I),I=1,2)/'STANDARDS','UNKNOWNS'/
255       DATA (Pstr(30,I),I=3,6)/'DIRECT','LINEAR','DIRECT-QUAD',
256      &'QUADRATIC'/
257       DATA (Pstr(31,I),I=7,8)/'NORMAL','AVERAGED'/
258       DATA (Pstr(32,I),I=9,13)/'SIGNAL AV','SAMPLE AV','QUICK',
259      &'EXTENDED','FIXED'/
260       DATA (Pstr(33,I),I=1,5)/'DSPL RESULTS','DSPL SETUP','NEXT CONC',
261      &'DELETE SAMPLE','CLEAR RESULTS'/
262       DATA (Pstr(34,I),I=1,2)/'OFF','ON'/
263       DATA (Pstr(35,I),I=1,2)/'1','2'/
264       DATA (Pstr(38,I),I=1,5)/'  OFF','  ON','RECORD',' ','ON/SETUP'/
265       DATA (Pstr(40,I),I=1,2)/'INTERVAL','ACCY-DRIVEN'/
```

```
C
      DATA (Fit(I),I=1,5)/' 7','11','15','21','25'/
C
      Pfit=' POINT FIT'
      BELL=CHAR(7)
      ESC=CHAR(27)
      CLR=ESC//'J'
      HOME=ESC//'h'
      UP=ESC//'A'
      DOWN=ESC//'B'
      ERASE=ESC//'K'
C
C     ----------------------------------------------------------------
C
C         Assign Statement Labels
C
C     ----------------------------------------------------------------
C
      ASSIGN 10  TO MENU
      ASSIGN 100 TO READ
      ASSIGN 300 TO SETTINGS
      ASSIGN 400 TO EDIT
      ASSIGN 1000 TO STORE
      ASSIGN 2000 TO CURFILE
      ASSIGN 9999 TO EXIT
C
C     ----------------------------------------------------------------
C
C         Data Transfer And Edit Control Menu
C
C     ----------------------------------------------------------------
C
      CALL FFRCL(79)                          ! Eliminate Line Wrap
   10 WRITE (1,*) HOME,CLR,'_'
      NCOL=70
      TITLE='Cary 2390'
      CALL Center(TITLE)
      TITLE='Spectral Data Editing'
      CALL Center(TITLE)
      WRITE (1,'(T61,A2,A8)') UP,'Rev: 1.8'
      CALL Line(NCOL)
      WRITE (1,20) 'CODE','FUNCTION','STATUS','MIN','MAX','INC'
   20 FORMAT (T4,A4,T18,A8,T38,A7,T50,A3,T58,A3,T66,A3)
      CALL Line(NCOL)
      WRITE (1,30) 'R','.....Read Spectrum..........',Sstat(1),
     &Smin,Smax,Sinc
      WRITE (1,40) 'I','.....Instrument Setti..gs.....',Sname(1)
      WRITE (1,30) 'E','.....Edit Spectrum..........',Fname,
     &Emin,Emax,Einc
      WRITE (1,50) 'S','.....Store Spectrum.........',Fstat
      WRITE (1,50) 'D','.....Store Derivative.......',Dstat
      WRITE (1,50) 'C','.....Store Curve Fit File....',Cstat
      WRITE (1,60) 'X','.....EXIT Menu.............'
```

```
319     30 FORMAT (/,T6,A,T7,A30,T39,A10,T50,A4,T58,A4,T66,A4)
320     40 FORMAT (/,T6,A,T7,A30,T39,A20)
321     50 FORMAT (/,T6,A,T7,A30,T39,A10)
322     60 FORMAT (/,T6,A,T7,A30)
323        WRITE (1,*)
324        CALL Line(NCOL)
325        WRITE (1,*)
326     70 WRITE (1,*) UP,ERASE,'_'
327        WRITE (1,'(T3,A15,A,A2)') 'Enter the CODE:',BELL,'_'
328        READ (1,80) Code
329        CALL Upper(Code)
330        IF (Code.EQ.'R') GO TO READ
331        IF (Code.EQ.'I') GO TO SETTINGS
332        IF (Code.EQ.'E') GO TO EDIT
333        IF (Code.EQ.'S') GO TO STORE
334        IF (Code.EQ.'D') GO TO STORE
335        IF (Code.EQ.'C') GO TO CURFILE
336        IF (Code.EQ.'X') GO TO EXIT
337        GO TO 70
338     80 FORMAT (A1)
339     90 FORMAT (A14)
340 C
341 C      ------------------------------------------------------------
342 C
343 C          Read Spectrum
344 C
345 C      ------------------------------------------------------------
346 C
347    100 TITLE='Read Spectrum'
348        J=1
349        IF (Sstat(1).NE.'VALID') GO TO 120
350    110 WRITE (1,*) UP,ERASE,'  Spectrum is PRESENT: ',
351       &'N...New Spectrum, B...Baseline, Q...Quit ? ',BELL,'_'
352        READ (1,80) Icode
353        CALL Upper(Icode)
354        IF (Icode.EQ.'Q') GO TO 70
355        IF (Icode.EQ.'N') GO TO 120
356        IF (Icode.NE.'B') GO TO 110
357        IF (Sstat(2).EQ.'VALID') THEN
358          WRITE (1,*) UP,ERASE,'  Baseline SUBTRACTED: ',BELL
359          CALL Wait (2.0)
360          GO TO 110
361        END IF
362        TITLE='Subtract Baseline'
363        J=2
364    120 WRITE (1,*) HOME,CLR,'_'
365        CALL Center(TITLE)
366        CALL Line(NCOL)
367        WRITE (1,*) DOWN,'  Enter Filename: ',BELL,'_'
368        READ (1,'(A20)') Sname(J)
369        WRITE (1,*) DOWN,'  Directory, (RETURN = /DEFAULT/): ',BELL,'_'
370        READ (1,'(A40)') Directory
```

```
      IF (Directory.EQ.' ') THEN
        Filespec=Sname(J)
        GO TO 130
      END IF
      K=40
      DO WHILE (Directory(K:K).EQ.' ')
        K=K-1
      END DO
      Filespec=Directory(1:K)//'/'//Sname(J)
  130 K=63
      DO WHILE (Filespec(K:K).EQ.' ')
        K=K-1
      END DO
      WRITE (1,*) UP,ERASE,'  Validating: ',Filespec(1:K),' _'
      CALL Wait(1.0)
      OPEN (UNIT=66,FILE=Filespec(1:K),IOSTAT=N,STATUS='OLD')
      WRITE (1,*)
      IF (N.NE.0) THEN
        N=N-500
        WRITE (1,*) UP,ERASE,'_'
        IF (N.EQ.6) WRITE (1,*) ' File does NOT EXIST: ',BELL,'_'
        IF (N.EQ.8) WRITE (1,*) ' File is already OPEN: ',BELL,'_'
        IF ((N.NE.6).AND.(N.NE.8)) WRITE (1,*) ' Disk Error # ',N,
     &  BELL,'_'
        CALL Wait(2.0)
        WRITE (1,*)
        IF (J.EQ.1) Sstat(J)='INVALID'
        GO TO MENU
      END IF
      Fname=' '                    ! New File Opened So Erase Any
      Emin=' '                     ! Previous Descriptors For
      Emax=' '                     ! A Valid Edited Spectrum
      Einc=' '
      Sstat(2)=' '
      WRITE  (1,*) UP,ERASE,'  Reading File: ',Filespec(1:K),BELL
      READ (66,FMT=140,IOSTAT=N,ERR=230) LABEL(J)
  140 FORMAT (A72)
      WRITE (1,*) DOWN,'  Title: '
      WRITE (1,*) '  ',LABEL(J)
      READ (66,FMT=150,IOSTAT=N,ERR=230) DATE(J)
  150 FORMAT (A8)
      WRITE (1,*) DOWN,'  Date: ',DATE(J)
      READ (66,*,IOSTAT=N,ERR=230) XMIN(J),XMAX(J),XSTEP(J),CONC(J),
     &PATH(J)
      READ (66,*,IOSTAT=N,ERR=230) ORD(J),ABSC(J),CELL(J),CYCLE(J),
     &SAMPLE(J),WAVE(J),TIMER(J),TEMP(J),DIST(J)
      WRITE (1,160) DOWN,'Scan Range: ',XMAX(J),' to ',XMIN(J),
     &' nm at ',XSTEP(J),' nm steps'
      WRITE (1,'(T4,A2,A12,2X,G9.4)') DOWN,'Conc. (M) : ',CONC(J)
      WRITE (1,170) DOWN,'Path (cm) : ',PATH(J)
      WRITE (1,170) DOWN,'Temp. (C) : ',TEMP(J)
  160 FORMAT (T4,A2,A12,F6.2,A4,F6.2,A7,F4.2,A9)
  170 FORMAT (T4,A2,A12,F6.2)
```

```
      READ (66,FMT=180,IOSTAT=N,ERR=230) NP,NV,ND(J)
  180 FORMAT (I3,I3,I6)
      IF ((NP.NE.49).OR.(NV.NE.14)) GO TO 220     ! Data Format Error
      WRITE (1,*)
      IF (ABS(XMIN(J)-WAVE(J)).GT.0.2) THEN
        WRITE (1,*) '  Scan ended at',WAVE(J),' nm',BELL
        CALL Wait(2.0)
        WRITE (1,*) UP,ERASE,UP,UP
      END IF
      WRITE (1,*) HOME,'_'
      TITLE='### READING DATAFILE ###'
      CALL Center(TITLE)
      READ (66,FMT=190,IOSTAT=N,ERR=230) (PARAM(J,K),K=1,NP)
  190 FORMAT (I2)
      READ (66,*,IOSTAT=N,ERR=230) (VARIABLE(J,K),K=1,NV)
      NDATA=ND(1)
      IF (J.EQ.1) THEN
        READ (66,*,IOSTAT=N,ERR=230) (Y(I),I=1,NDATA)
        READ (66,*,IOSTAT=N,ERR=230) (X(I),I=1,NDATA)
        WMIN=XMIN(J)
        WMAX=XMAX(J)
        STEP=XSTEP(J)
        CALL Str(WMIN,String,4)
        Smin=String(2:5)
        CALL Str(WMAX,String,4)
        Smax=String(2:5)
        CALL Str(STEP,String,4)
        Sinc=String(2:5)
        Sstat(J)='VALID'
        IF (PARAM(1,1).EQ.0) THEN
          Ymode='A'                ! Ordinate Mode = ABSORBANCE
        ELSE
          Ymode='T'                ! Ordinate Mode = TRANSMISSION
        END IF
        GO TO 210                  ! Close Spectrum File
      END IF
C     ------------------------------------------------------------
C         Test For Baseline Matching
C     ------------------------------------------------------------
      MATCH=.TRUE.
      IF (XMIN(1).LT.XMIN(2)) MATCH=.FALSE.
      IF (XMAX(1).GT.XMAX(2)) MATCH=.FALSE.
      IF (XSTEP(1).NE.XSTEP(2)) MATCH=.FALSE.
      IF (PARAM(1,1).NE.PARAM(2,1)) MATCH=.FALSE.
      IF (PARAM(1,5).NE.PARAM(2,5)) MATCH=.FALSE.
      IF (PARAM(1,16).NE.PARAM(2,16)) MATCH=.FALSE.
      IF (PARAM(1,21).NE.PARAM(2,21)) MATCH=.FALSE.
      IF (PARAM(1,22).NE.PARAM(2,22)) MATCH=.FALSE.
      IF (PARAM(1,23).NE.PARAM(2,23)) MATCH=.FALSE.
      IF (PARAM(1,38).NE.PARAM(2,38)) MATCH=.FALSE.
      IF (.NOT.MATCH) GO TO 240
C     ------------------------------------------------------------
C         Read Entire Baseline File
```

```
C     ------------------------------------------------------------
      READ (66,*,IOSTAT=N,ERR=230) (YB(I), I=1,ND(2))
      READ (66,*,IOSTAT=N,ERR=230) (XB(I), I=1,ND(2))
C     ------------------------------------------------------------
C        Match Baseline Index To Starting Wavelength Of Spectrum
C     ------------------------------------------------------------
      START=0
      IF (WMAX.LT.XMAX(2)) THEN
        START=NINT((XMAX(2)-WMAX)/XSTEP(2))
      END IF
      DO 200 I=1,NDATA                  ! Subtract Relevant Wavelength
        K=I+START                       ! Region Of Baseline File :-
        IF (Ymode.EQ.'A') THEN          ! Absorbance Mode
          Y(I)=Y(I)-YB(K)               !   SUBTRACT BASELINE DATA
        ELSE                            ! Transmission Mode
          Y(I)=(Y(I)/YB(K))*100.0       !   RATIO TO BASELINE DATA
        END IF                          ! NB: You Can't READ & SUBTRACT
  200 CONTINUE                          ! With The Implied DO LOOP FORM
      Sstat(J)='VALID'                  ! Set Baseline To VALID
  210 CLOSE (UNIT=66,IOSTAT=N,ERR=230)
      COUNT=0                           ! Zero Smoothing Cycle Counter
      J=1                               ! Select Spectrum Mode Before
      GO TO MENU                        ! Returning To Main Menu
C     ------------------------------------------------------------
  220 WRITE (1,*) UP,ERASE,'  Data Format Error: ',BELL,'_'
      Sstat(J)='ABORTED'
      GO TO 250
  230 WRITE (1,*) UP,ERASE,'  Disk Error #',N,BELL,' _'
      Sstat(J)='INVALID'
      GO TO 250
  240 WRITE (1,*) UP,ERASE,'  Baseline Mismatch Error: ',BELL,'_'
  250 CALL Wait(2.0)
      WRITE (1,*)
      GO TO 210
  260 WRITE (1,*) UP,ERASE,'  Spectrum is ABSENT: ',BELL,'_'
      CALL Wait(2.0)
      WRITE (1,*)
      GO TO 70
C
C     ------------------------------------------------------------
C
C        Display Instrument Settings
C
C     ------------------------------------------------------------
C
  300 IF (Sstat(1).NE.'VALID') GO TO 260
      IF (Sstat(2).EQ.'VALID') THEN
        WRITE (1,*) UP,ERASE,'  S...Spectrum or B...Baseline ? ',
     &  BELL,'_'
        READ (1,80) Icode
        CALL Upper(Icode)
        J=1
        IF (Icode.EQ.'B') J=2
```

```
530       END IF
531       BAND=VARIABLE(J,10)
532       GAIN=VARIABLE(J,6)
533       CALL Val(Pstr(7,PARAM(J,7)+1),PMAX)
534       PMIN=VARIABLE(J,1)
535       IF (PARAM(J,1).NE.0) THEN
536         CALL Val(Pstr(8,PARAM(J,8)+1),PMAX)
537         PMIN=VARIABLE(J,11)
538       END IF
539       PMAX=PMIN+PMAX
540       K=11
541       IF (PARAM(J,6).EQ.4) K=10
542       String=Pstr(K,PARAM(J,K)+1)
543       WRITE (1,*) HOME,CLR,'_'
544       TITLE='Instrument Settings'
545       CALL Center(TITLE)
546       CALL Line(50)
547       WRITE (1,'(T20,A8,T40,A7)') 'FUNCTION','SETTING'
548       CALL LINE(50)
549       WRITE (1,*)
550       WRITE (1,320) 'WAVELENGTH LIMITS.....',
551      &XMIN(J),',',XMAX(J)
552       WRITE (1,330) 'ORDINATE.............',
553      &Pstr(1,PARAM(J,1)+1)
554       WRITE (1,330) 'ABSCISSA.............',
555      &Pstr(2,PARAM(J,2)+1)
556       WRITE (1,330) 'SCAN RATE (nm/sec).....',
557      &Pstr(3,PARAM(J,3)+1)
558       WRITE (1,330) 'CHART DISPLAY (nm/cm)..',
559      &Pstr(4,PARAM(J,4)+1)
560       WRITE (1,330) 'REFERENCE MODE.........',
561      &Pstr(5,PARAM(J,5)+1)
562       WRITE (1,340) 'SBW (nm), GAIN.........',
563      &BAND,',',GAIN
564       WRITE (1,330) 'PEN FUNCTION...........',
565      &Pstr(6,PARAM(J,6)+1)
566       WRITE (1,340) 'PEN LIMITS (Min,Max)...',
567      &PMIN,',',PMAX
568       IF (PARAM(J,6).GT.1) WRITE (1,350) UP,ERASE,String
569       WRITE (1,330) 'RESPONSE TIME (sec)....',
570      &Pstr(15,PARAM(J,15)+1)
571       WRITE (1,330) 'BEAM INTERCHANGE.......',
572      &Pstr(16,PARAM(J,16)+1)
573       WRITE (1,330) 'SLIT HEIGHT............',
574      &Pstr(23,PARAM(J,23)+1)
575       WRITE (1,330) 'LAMP SELECT............',
576      &Pstr(21,PARAM(J,21)+1)
577       WRITE (1,330) 'DETECTOR SELECT........',
578      &Pstr(22,PARAM(J,22)+1)
579   320 FORMAT (T15,A23,T40,F5.2,A,F5.2)
580   330 FORMAT (T15,A23,T40,A14)
581   340 FORMAT (T15,A23,T40,F4.2,A,F5.2)
582   350 FORMAT (T40,A2,A2,A14)
```

```
583         WRITE (1,*)
584         CALL Line(50)
585         WRITE (1,*)
586      360 WRITE (1,'(T14,A2,A2,A23,A,A)') UP,ERASE,
587        &'Press RETURN for MENU: ',BELL,'_'
588         READ (1,80) Icode
589         CALL Upper(Icode)
590         IF (Icode.NE.' ') GO TO 360
591         GO TO MENU
592 C
593 C       ---------------------------------------------------------------
594 C
595 C           Edit Spectral File Parameters
596 C
597 C       ---------------------------------------------------------------
598 C
599      400 IF (Sstat(1).NE.'VALID') GO TO 260
600         WRITE (1,*) HOME,CLR,'_'
601         TITLE='Edit Spectral Parameters'
602         CALL Center(TITLE)
603         CALL Line(40)
604         WRITE (1,*)
605         WRITE (1,410) 'C .....Concentration, (M)'
606         WRITE (1,410) 'D .....Date, (MM/DD/YY)  '
607         WRITE (1,410) 'L .....Label, (72 chars) '
608         WRITE (1,410) 'P .....Pathlength, (cm)  '
609         WRITE (1,410) 'S .....Spectral Data     '
610         WRITE (1,410) 'X .....EXIT to Menu      '
611      410 FORMAT (T24,A25,/)
612         CALL Line(40)
613         WRITE (1,*)
614      420 WRITE (1,430) UP,ERASE,'_'
615      430 FORMAT (T19,A2,A2,A)
616         WRITE (1,'(A17,A,A)') ' Enter the CODE: ',BELL,'_'
617         READ (1,80) Icode
618         CALL Upper(Icode)
619         IF (Icode.EQ.'C') GO TO 440
620         IF (Icode.EQ.'D') GO TO 460
621         IF (Icode.EQ.'L') GO TO 470
622         IF (Icode.EQ.'P') GO TO 480
623         IF (Icode.EQ.'S') GO TO 500
624         IF (Icode.EQ.'X') GO TO MENU
625         GO TO 420
626 C       ---------------------------------------------------------------
627      440 WRITE (1,430) UP,ERASE,'_'
628         WRITE (1,*) 'Concentration,(M):  ',CONC(1)
629         WRITE (1,450) UP,BELL
630      450 FORMAT (T39,A2,A,'? _')
631         READ (1,90) String
632         IF (String.EQ.' ') GO TO 420
633         CALL Val(String,CONC(1))
634         CONC(1)=ABS(CONC(1))
635         GO TO 420
```

```
C     ------------------------------------------------------------------
  460 WRITE (1,430) UP,ERASE,'_'
      WRITE (1,*) 'Date, (MM/DD/YY):     ',DATE(1)
      WRITE (1,450) UP,BELL
      READ (1,150) String
      IF (String.EQ.' ') GO TO 420
      DATE(1)=String(1:8)
      GO TO 420
C     ------------------------------------------------------------------
  470 WRITE (1,*) UP,ERASE,'Label:',DOWN
      WRITE (1,'(A3,A72)') '   ',LABEL(1)
      WRITE (1,'(T4,A,T75,A)') '↑','↑'
      WRITE (1,*) UP,UP,BELL,'? _'
      READ (1,140) TITLE
      WRITE (1,*) ERASE,UP,ERASE,UP,UP,ERASE
      IF (TITLE.EQ.' ') GO TO 420
      LABEL(1)=TITLE
      GO TO 420
C     ------------------------------------------------------------------
  480 WRITE (1,430) UP,ERASE,'_'
      WRITE (1,490) 'Pathlength, (cm):   ',PATH(1)
  490 FORMAT (X,A20,X,F5.4)
      WRITE (1,450) UP,BELL
      READ (1,90) String
      IF (String.EQ.' ') GO TO 420
      CALL Val(String,PATH(1))
      PATH(1)=ABS(PATH(1))
      GO TO 420
C     ------------------------------------------------------------------
  500 TITLE='Edit Spectral Data'
      WRITE (1,*) HOME,CLR,'_'
      CALL Center(TITLE)
      CALL Line(40)
      WRITE (1,*)
      WRITE (1,510) 'L .....List Data S. ment'
      WRITE (1,510) 'E .....Edit Single Datum'
      WRITE (1,510) 'O .....Offset Segment   '
      WRITE (1,510) 'N .....Near IR Rescale  '
      WRITE (1,510) 'R .....Reduce File Size '
      WRITE (1,510) 'S .....Smooth Segment   '
      WRITE (1,510) 'D .....Deriv. Spectra   '
      WRITE (1,510) 'X .....EXIT Data Editing'
  510 FORMAT (T26,A24,/)
      CALL Line(40)
      WRITE (1,*)
  520 WRITE (1,530) UP,ERASE,'_'
  530 FORMAT (T19,A2,A2,A)
      WRITE (1,540) ' Enter the CODE: ',BELL,'_'
  540 FORMAT (A17,A,A)
      READ (1,80) Code
      CALL Upper(Code)
      WRITE (1,*) UP,ERASE,UP
```

```
688         IF (Code.EQ.'L') GO TO 550
689         IF (Code.EQ.'E') GO TO 630
690         IF (Code.EQ.'O') GO TO 660
691         IF (Code.EQ.'N') GO TO 660
692         IF (Code.EQ.'R') GO TO 660
693         IF (Code.EQ.'S') GO TO 740
694         IF (Code.EQ.'D') GO TO 740
695         IF (Code.NE.'X') THEN
696           WRITE (1,*)
697           GO TO 520
698         END IF
699         CALL Str(WMIN,String,4)
700         Emin=String(2:5)
701         CALL Str(WMAX,String,4)
702         Emax=String(2:5)
703         CALL Str(STEP,String,4)
704         Einc=String(2:5)
705         Fname='VALID'
706         GO TO 400
707 C       ---------------------------------------------------------------
708    550  WRITE (1,*) '  Scan Range: ',WMIN,' to',WMAX,' nm at',
709        &STEP,' nm steps',DOWN,DOWN
710    560  WRITE (1,*) UP,ERASE,'  DISPLAY Range: (MIN,MAX) ? ',BELL,'_'
711         READ (1,*,ERR=560) WN,WX
712         WN=ABS(WN)
713         WX=ABS(WX)
714         CALL Xlimits(WN,WX)
715         IF ((WN.LT.WMIN).OR.(WN.GT.WMAX)) GO TO 560
716         IF ((WX.LT.WMIN).OR.(WX.GT.WMAX)) GO TO 560
717         START=NINT((WMAX-WX)/STEP)+1
718         FINISH=NINT(NDATA-(WN-WMIN)/STEP)
719         IF (Ymode.EQ.'A') THEN
720           TITLE='Wavelength    Absorbance    Ext.Coeff. (/M/cm)'
721         ELSE
722           TITLE='Wavelength         % Transmission'
723         END IF
724         WRITE (1,*) HOME,CLR,'_'
725         CALL Center(TITLE)
726         CALL Line(50)
727         L=1
728         DO 580 I=START,FINISH
729           WL=(WMAX-FLOAT(I-1)*STEP)          ! Calculated Wavelength
730           EC=Y(I)/CONC(1)/PATH(1)            ! Extinction Coefficient
731           WRITE (1,*)
732           IF (Ymode.EQ.'A') THEN
733             WRITE (1,590) X(I),Y(I),EC
734           ELSE
735             WRITE (1,600) X(I),Y(I)
736           END IF
737           L=L+1
738           IF (L.EQ.10) THEN
739             WRITE (1,*) DOWN
740    570      WRITE (1,610) UP,ERASE,'Press RETURN to Continue ',BELL,'_'
```

```
741           READ (1,80) Icode
742           CALL Upper(Icode)
743           IF (Icode.NE.' ') GO TO 570
744           WRITE (1,*) HOME,CLR,'_'
745           CALL Center(TITLE)
746           CALL Line(50)
747           L=1
748         END IF
749   580 CONTINUE
750   590 FORMAT (T16,F7.2,T30,F7.4,T45,G10.4)
751   600 FORMAT (T22,F7.2,T41,F7.2)
752   610 FORMAT (T15,A2,A2,A25,A,A)
753       WRITE (1,*) DOWN
754   620 WRITE (1,610) UP,ERASE,'Press RETURN for MENU    ',BELL,'_'
755       READ (1,80) Icode
756       CALL Upper(Icode)
757       IF (Icode.NE.' ') GO TO 620
758       GO TO 500
759 C     ------------------------------------------------------------------
760   630 WRITE (1,*) '  Scan Range: ',WMIN,' to',WMAX,' nm at',
761      &STEP,' nm steps',DOWN,DOWN
762   640 WRITE (1,*) UP,ERASE,'  EDIT Wavelength: ? ',BELL,'_'
763       READ (1,*,ERR=640) WL
764       WL=ABS(WL)
765       IF ((WL.LT.WMIN).OR.(WL.GT.WMAX)) GO TO 640
766       I=NINT((WMAX-WL)/STEP)+1
767       WRITE (1,*) UP,ERASE,UP,UP,ERASE,'   Wavelength: ',X(I),' nm _'
768       WRITE (1,*) '(Expected:',(WMAX-FLOAT(I-1)*STEP),' nm)'
769       IF (Ymode.EQ.'A') THEN
770         String='ABSORBANCE:  _'
771       ELSE
772         String='% T or % R:  _'
773       END IF
774       WRITE (1,'(T5,A2,A14)') DOWN,String
775       WRITE (1,*) Y(I)
776   650 WRITE (1,'(T18,A2,A,A3)') UP,BELL,'? _'
777       READ (1,90) String
778       IF (String.EQ.' ') GO TO 500
779       CALL Val(String,NUMBER)
780       IF (Ymode.EQ.'A') THEN
781         IF ((NUMBER.LT.-0.5).OR.(NUMBER.GT.4.5)) GO TO 650
782       ELSE
783         IF ((NUMBER.LT.0.0).OR.(NUMBER.GT.200.0)) GO TO 650
784       END IF
785       Y(I)=NUMBER
786       GO TO 500
787 C     ------------------------------------------------------------------
788   660 WRITE (1,*) '  Scan Range: ',WMIN,' to',WMAX,' nm at',
789      &STEP,' nm steps',DOWN,DOWN
790       String='OFFSET'
791       K=6
792       IF (Code.EQ.'N') GO TO 700
```

```
      IF (Code.EQ.'R') THEN
        String='FILE'
        K=4
      END IF
  670 WRITE (1,*) UP,ERASE,'  ',String(1:K),' Range: (MIN,MAX) ? ',
     &BELL,'_'
      READ (1,*,ERR=670) WN,WX
      WN=ABS(WN)
      WX=ABS(WX)
      CALL Xlimits(WN,WX)
      IF ((WN.LT.WMIN).OR.(WN.GT.WMAX)) GO TO 670
      IF ((WX.LT.WMIN).OR.(WX.GT.WMAX)) GO TO 670
      START=NINT((WMAX-WX)/STEP)+1
      FINISH=NINT(NDATA-(WN-WMIN)/STEP)
      IS=1
C     ------------------------------------------------------------------
      IF (Code.EQ.'O') GO TO 700
  680 WRITE (1,*) UP,ERASE,'  STEP Multiplier: ? ',IS
      WRITE (1,'(T24,A2,A,A)') UP,BELL,'_'
      READ (1,90) String
      IF (String.NE.' ') THEN
        CALL Val(String,NUMBER)
        IS=NINT(NUMBER)
        IF (IS.GE.NDATA) GO TO 680
      END IF
      J=1
      DO 690 I=START,FINISH,IS
        X(J)=X(I)                          ! Reduced File Begins At
        Y(J)=Y(I)                          ! INDEX=1 - Old File Data
        J=J+1                              ! Is Written Over
  690 CONTINUE
      WAVE=X(FINISH)                       ! Update File Parameters
      ABSC=WAVE                            ! To Match Reduced File
      NDATA=(FINISH-START)/IS+1
      ND(1)=NDATA
      STEP=STEP*FLOAT(IS)
      WMAX=ANINT(X(1))
      WMIN=ANINT(X(NDATA))
      GO TO 500
C     ------------------------------------------------------------------
  700 YOFF=0.0
      IF (Code.EQ.'O') THEN
        String='Absorbance'
        IF (Ymode.EQ.'T') THEN
          String='% T or % R'
        END IF
        GO TO 710
      END IF
      IF ((PARAM(1,5).NE.0).OR.(WMIN.GE.800.0).OR.(WMAX.LE.800.0)) THEN
        WRITE (1,*) UP,ERASE,'  NIR/VIS Change Not Used: ',BELL,'_'
        CALL Wait(2.0)
        GO TO 500
      END IF
```

```
      String='NIR Region'
      START=1
      FINISH=NINT((WMAX-800.0)/STEP)+1
      YOFF=Y(FINISH+1)-Y(FINISH)  ! Offset At 800 nm Detector Change
  710 WRITE (1,720) UP,ERASE,String(1:10),' OFFSET:    ',YOFF
  720 FORMAT (T4,A2,A2,A10,A11,F7.4)
      WRITE (1,'(T24,A2,A,A3)') UP,BELL,'? _'
      READ (1,90) String
      IF (String.NE.' ') THEN
        CALL Val(String,YOFF)
      END IF
      IF (YOFF.EQ.0.0) GO TO 500
      IF (Code.EQ.'N') THEN
        Y2=Y(FINISH+1)              ! First Datum In Visible Region
        Y1=Y2-YOFF                  ! Final Datum In Near IR Region
      END IF
      DO 730 I=START,FINISH
        IF (Code.EQ.'O') Y(I)=Y(I)+YOFF
        IF (Code.EQ.'N') Y(I)=Y(I)*Y2/Y1
  730 CONTINUE
      GO TO 500
C
C     ---------------------------------------------------------------
C
C        Smooth Spectrum
C
C     ---------------------------------------------------------------
C
  740 TITLE='Smooth Spectrum'
      DERIV=.FALSE.
      IF (Code.EQ.'D') GO TO 880      ! Transfer To Derivative Setup
C     ---------------------------------------------------------------
  750 WRITE (1,*) HOME,CLR,'_'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,*) DOWN,'  Select ALGORITHM:',DOWN
      DO 760 I=1,5
        WRITE (1,770) I,'.....',Fit(I),Pfit
  760 CONTINUE
  770 FORMAT (T25,I1,X,A5,X,A2,A10,/)
      I=6
      WRITE (1,'(T25,I1,X,A18)') I,'..... EXIT to Menu'
      WRITE (1,*) DOWN
  780 WRITE (1,790) UP,ERASE,'Enter the CODE # :  ',BELL,'_'
  790 FORMAT (T5,A2,A2,A20,A,A)
      READ (1,80) Icode
      L=ICHAR(Icode)-48
      IF ((L.LT.1).OR.(L.GT.6)) GO TO 780
      IF (L.EQ.6) GO TO 500
      WRITE (1,*) UP,ERASE,UP,UP,ERASE,UP,UP,ERASE,UP,UP,ERASE,UP,UP,
     &ERASE,UP,UP,ERASE,UP,UP,ERASE,UP,UP,ERASE
      WRITE (1,*) UP,'  ALGORITHM: ',Fit(L),Pfit,' (Quartic)'
```

```
C     ------------------------------------------------------------
C
C         Step Multiple Defaults To 2 For Improved Smoothing:
C
C         - The Smoothing Routine Increments 1 Datum Per Fit.
C         - If The Noise Is Non-Random, Increasing The Step Size May
C           Produce A Sinusoidal Approximation, Which Creates Very
C           Poor Second Derivative Spectra. Smooth Those Regions With
C           A Large Step Size To Average The Noise And Then With The
C           Step Size Reduced To 1 Datum To Eliminate Any Ripples.
C         - In General, Higher Multiples Of The Data Step Size Produce
C           Better Smoothing & MUCH BETTER Derivatives.
C
C     ------------------------------------------------------------
      IS=2
      RANGE=FLOAT(NPTS(L)-1)*STEP*FLOAT(IS)
      WRITE (1,*) DOWN,'  Default SEGMENT Size:',RANGE,' nm'
      WRITE (1,*) DOWN
      WRITE (1,*) UP,ERASE,'_'
      WRITE (1,'(X,A20,I1)') ' STEP Multiplier:    ',IS
      WRITE (1,'(T21,A2,A,A3)') UP,BELL,'? _'
      READ (1,90) String
      IF (String.EQ.' ') GO TO 800
      CALL Val(String,NUMBER)
      IS=NINT(NUMBER)
      IF (IS.LT.1) IS=1
      ITEST=(NDATA-1)/(NPTS(L)-1)
      IF (IS.GT.ITEST) IS=ITEST
  800 RANGE=FLOAT(NPTS(L)-1)*STEP*FLOAT(IS)
      WRITE (1,*) UP,ERASE,UP,UP,ERASE,'  Fitted SEGMENT Size:',
     &RANGE,' nm'
C
C     ------------------------------------------------------------
C
C                          LEAST SQUARES SMOOTHING
C                              OF DATA SEGMENT
C
C                          QUARTIC POLYNOMIAL FITS
C
C         X-INDEX:  -XX,...,0,...,+XX Allows For Simplification Of
C                   Least Squares Summations - Odd Powers = Zero
C
C         OFFSETS:  The Algorithm Can NOT Smooth The First Or Last
C                   (XX*IS) Number Of Points In The Spectrum
C
C         X-STEPS:  The Effectiveness Of The Smoothing Can Be Improved
C                   By Widening The Wavelength Interval Used By The
C                   Routine (Range =  IS*XX*STEP nm), Reducing The
C                   Spectral Resolution. The User Must Judge Which
C                   Algorithm & Step Multiplier (IS) Is Best Suited
C                   For The Particular Spectrum. The Minimum Step
C                   Size Is 1 Datum. This Is Adequate For Smoothing
C                   But NOT For Derivative Spectra.
```

```
951 C
952 C     ---------------------------------------------------------------
953 C
954       WOFF=STEP*FLOAT(IS*(NPTS(L)-1)/2)          ! Wavelength Offset
955       XOFF=(NPTS(L)+1)/2                         ! X-Index Offset+1
956 C     ---------------------------------------------------------------
957       WRITE (1,*) DOWN,'  Scan Range: ',WMIN,' to',WMAX,' nm at',
958      &STEP,' nm steps'
959       WRITE (1,*) DOWN
960       IF (DERIV) GO TO 910              ! Re-enter Derivative Routine
961 C     ---------------------------------------------------------------
962   810 WRITE (1,*) UP,ERASE,'  SMOOTHING Range: (MIN,MAX) ',BELL,'_'
963       READ (1,*,ERR=810) WN,WX
964       WN=ABS(WN)
965       WX=ABS(WX)
966       CALL Xlimits(WN,WX)
967       IF (WN.LT.WMIN+WOFF) WN=WMIN+WOFF
968       IF (WX.GT.WMAX-WOFF) WX=WMAX-WOFF
969       START=NINT((WMAX-WX)/STEP)+1
970       FINISH=NINT(NDATA-(WN-WMIN)/STEP)
971       WRITE (1,*) UP,ERASE,'  SMOOTHING Range: ',WN,' to',WX,' nm',
972      &DOWN,DOWN
973   820 WRITE (1,*) UP,ERASE,'  Total # of Smoothing PASSES: ',BELL,'_'
974       READ (1,*,ERR=820) NPASS
975       IF (NPASS.LT.1) GO TO 820
976 C     ---------------------------------------------------------------
977       WRITE (1,*) DOWN
978       DO 870 N=1,NPASS
979         WRITE (1,*) UP,ERASE,'  Pass #',N,' in PROGRESS:',BELL
980         DO 850 I=START,FINISH
981           DO 830 J=1,5
982             SS(J)=0.0                        ! Zero Summations Array
983   830       CONTINUE
984           DO 840 K=1,NPTS(L)
985             XX=K-XOFF                        ! Segment Is (-XX,+XX):
986             YY=DBLE(Y(I+XX*IS))              ! Step Size Is IS*STEP
987             SS(1)=SS(1)+YY                   ! Calculate Summations
988             SS(3)=SS(3)+YY*DBLE(XX**2)       ! For Least Squares Fit
989             SS(5)=SS(5) YY*DBLE(XX**4)       ! Within Segment:
990   840       CONTINUE
991 C     ---------------------------------------------------------------
992 C         Array CC Holds Known Least Squares Coefficients
993 C     ---------------------------------------------------------------
994           YY=CC(L,1)*SS(1)+CC(L,3)*SS(3)+CC(L,5)*SS(5)
995           Z(I)=SNGL(YY)                      ! Z(I) Is Smoothed Segment
996   850     CONTINUE                           ! For Range Requested Only
997         DO 860 I=START,FINISH
998           Y(I)=Z(I)                          ! Update Y(I) With Z(I)
999   860     CONTINUE
1000        COUNT=COUNT+1                        ! Update Total # Of Passes
1001  870 CONTINUE
1002      WRITE (1,*) UP,ERASE,'  Smoothing FINISHED: ',BELL,'_'
1003      CALL Wait(2.0)
```

```
1004        Fstat='SMOOTHED'
1005        IF (Code.EQ.'D') GO TO 900
1006        GO TO 500
1007 C
1008 C      ---------------------------------------------------------------
1009 C
1010 C          Derivative Spectrum Routine
1011 C
1012 C      ---------------------------------------------------------------
1013 C
1014    880 WRITE (1,*) HOME,CLR,'_'
1015        TITLE='Derivative Spectrum'
1016        CALL Center(TITLE)
1017        CALL Line(NCOL)
1018        WRITE (1,*) DOWN
1019    890 WRITE (1,*) UP,ERASE,'  SMOOTH Spectrum First, (Y or N) ? ',
1020       &BELL,'_'
1021        READ (1,80) Icode
1022        CALL Upper(Icode)
1023        IF (Icode.EQ.'Y') THEN
1024          DERIV=.FALSE.
1025          TITLE='Smooth Spectrum'
1026          GO TO 750
1027        END IF
1028        IF (Icode.NE.'N') GO TO 890
1029 C      ---------------------------------------------------------------
1030 C          1st Re-entry Point After Performing Smoothing Algorithm
1031 C      ---------------------------------------------------------------
1032    900 DERIV=.TRUE.
1033        TITLE='Derivative Spectrum'
1034        GO TO 750
1035 C      ---------------------------------------------------------------
1036 C          2nd Re-entry Point After Selecting Derivative Algorithm
1037 C      ---------------------------------------------------------------
1038    910 WN=WMIN+WOFF                        ; Set Limits For Calculation
1039        WX=WMAX-WOFF                        ! Of Derivative Spectrum
1040        START=NINT((WMAX-WX)/STEP)+1
1041        FINISH=NINT(NDATA-(WN-WMIN)/STEP)
1042    920 WRITE (1,*) UP,ERASE,'  FIRST or SECOND Derivative: (F or S) ?',
1043       &BELL,' _'
1044        READ (1,80) Icode
1045        CALL Upper(Icode)
1046        IF (Icode.EQ.'F') THEN
1047          String='FIRST'
1048          M=5
1049          ZM=10.0                             ! 1st Derivative Multiplier
1050          GO TO 930
1051        END IF
1052        IF (Icode.NE.'S') GO TO 920
1053        String='SECOND'
1054        M=6
1055        ZM=100.0                            ! 2nd Derivative Multiplier
1056 C      ---------------------------------------------------------------
```

```
930 WRITE (1,*) UP,ERASE,'  Calculating ',String(1:M),' Derivative:',
   &BELL
C     ------------------------------------------------------------------
      DO 970 I=START,FINISH
        DO 940 J=1,5
          SS(J)=0.0                          ! Zero Summations Array
940     CONTINUE
        DO 950 K=1,NPTS(L)
          XX=K-XOFF                          ! Segment Is (-XX,+XX):
          YY=DBLE(Y(I+XX*IS))                ! Step Size Is IS*STEP
          SS(1)=SS(1)+YY
          SS(2)=SS(2)+YY*DBLE(XX)            ! Calculate Summations
          SS(3)=SS(3)+YY*DBLE(XX**2)         ! For Least Squares Fit
          SS(4)=SS(4)+YY*DBLE(XX**3)         ! Within Segment:
          SS(5)=SS(5)+YY*DBLE(XX**4)
950     CONTINUE
C     ------------------------------------------------------------------
C        Array CC Holds Known Least Squares Coefficients
C     ------------------------------------------------------------------
        IF (Icode.EQ.'F') THEN
          YY=CC(L,2)*SS(2)+CC(L,4)*SS(4)
          GO TO 960
        END IF
        YY=2.0D0*(CC(L,3)*SS(1)+CC(L,6)*SS(3)+CC(L,7)*SS(5))
960     Z(I)=SNGL(YY)/STEP/FLOAT(IS)*ZM      ! Z(I) is Derivative Spectrum
970   CONTINUE                               ! Rescaled x10(F) or x100(S)
      DO 980 I=1,START-1
        Z(I)=Z(START)                        ! Set First XX*IS Points
980   CONTINUE
      DO 990 I=FINISH+1,NDATA
        Z(I)=Z(FINISH)                       ! Set Last XX*IS Points
990   CONTINUE
      WRITE (1,*) UP,ERASE,'  Derivative FINISHED: ',BELL,'_'
      CALL Wait(1.0)
      Dstat='VALID'
      Dtype=Icode
      DERIV=.FALSE.
      GO TO 500
C
C     ------------------------------------------------------------------
C
C        Store Edited Or Derivative Spectrum
C
C     ------------------------------------------------------------------
C
1000  IF (Sstat(1).NE.'VALID') GO TO 260
      IF ((Code.EQ.'D').AND.(Dstat.NE.'VALID')) GO TO 260
      IF (ABS(WMIN-WAVE(1)).GT.0.2) THEN
1010    WRITE (1,*) UP,ERASE,'  SCAN ENDED AT',WAVE(1),
     &  ' nm (Expected:',WMIN,'), Proceed (Y or N) ? ',BELL,'_'
        READ (1,80) Icode
        CALL Upper(Icode)
        IF (Icode.EQ.'N') GO TO 70
```

```
1110          IF (Icode.NE.'Y') GO TO 1010
1111        END IF
1112        HEADER=LABEL(1)
1113        IF (COUNT.GT.0) THEN
1114          NUMBER=FLOAT(COUNT)
1115          CALL Str(NUMBER,String,2)
1116          HEADER='SM'//String(2:3)//':'//LABEL(1)
1117        END IF
1118        String='Edited'
1119        M=6
1120        IF (Code.EQ.'D') THEN
1121          String='Derivative'
1122          M=10
1123          HEADER=Dtype//'D:'//LABEL(1)
1124        END IF
1125        TITLE='Store '//String(1:M)//' Spectrum'
1126  1020  WRITE (1,*) HOME,CLR,'_'
1127        CALL Center(TITLE)
1128        CALL Line(NCOL)
1129        WRITE (1,*)
1130        WRITE (1,*) '  Researcher''s Initials, (AA-ZZ): ',BELL,'_'
1131        READ (1,'(A2)') INITIALS
1132        WRITE (1,*) DOWN
1133        WRITE (1,*) UP,ERASE,'  Filename, (16 chars.): ',BELL,'_'
1134        READ (1,'(A16)') Fname
1135        L=16
1136        DO WHILE (Fname(L:L).EQ.' ')
1137          L=L-1
1138        END DO
1139        Directory=Fname(1:L)//'.S'//INITIALS
1140        Fname=Directory(1:L+4)
1141        WRITE (1,*) DOWN,'  Directory, (RETURN = /DEFAULT/): ',BELL,'_'
1142        READ (1,'(A40)') Directory
1143        IF (Directory.EQ.' ') THEN
1144          Filespec=Fname
1145          GO TO 1030
1146        END IF
1147        K=40
1148        DO WHILE (Directory(K:K).EQ.' ')
1149          K=K-1
1150        END DO
1151        IF (Directory(K:K).EQ.'/') K=K-1
1152        Filespec=Directory(1:K)//'/'//Fname
1153  1030  K=63
1154        DO WHILE (Filespec(K:K).EQ.' ')
1155          K=K-1
1156        END DO
1157        WRITE (1,*) UP,ERASE,'  Validating: ',Filespec(1:K),' _'
1158        OPEN (UNIT=66,FILE=Filespec(1:K),IOSTAT=N,STATUS='NEW')
1159        WRITE (1,*)
1160        IF (N.NE.0) THEN
1161          N=N-500
1162          WRITE (1,*) UP,ERASE,'_'
```

```
1163          IF (N.EQ.2) WRITE (1,*) ' FILE EXISTS: ',BELL,'_'
1164          IF (N.EQ.8) WRITE (1,*) ' FILE OPENED: ',BELL,'_'
1165          CLOSE (66,IOSTAT=N,ERR=1070)
1166          CALL Wait(2.0)
1167          GO TO 1020
1168        END IF
1169        IF (Icode.EQ.'R') GO TO 1050
1170        WRITE (1,*) DOWN,'  Title: '
1171        WRITE (1,*) '  ',HEADER
1172        WRITE (1,*) DOWN,'  Date: ',DATE(1)
1173        WRITE (1,*) DOWN,'  Concentration, (M): ',CONC(1)
1174        WRITE (1,*) DOWN,'  Pathlength, (cm): ',PATH(1)
1175 C      ------------------------------------------------------------
1176   1050 WRITE (1,*) DOWN,'  Storing File: ',Filespec(1:K),BELL
1177        WRITE (66,FMT=140,IOSTAT=N,ERR=1070) HEADER
1178        WRITE (66,FMT=150,IOSTAT=N,ERR=1070) DATE(1)
1179        WRITE (66,FMT=*,IOSTAT=N,ERR=1070) WMIN,WMAX,STEP,CONC(1),PATH(1)
1180        WRITE (66,FMT=*,IOSTAT=N,ERR=1070) ORD(1),ABSC(1),CELL(1),
1181       &CYCLE(1),SAMPLE(1),WAVE(1),TIMER(1),TEMP(1),DIST(1)
1182        WRITE (66,FMT=180,IOSTAT=N,ERR=1070) NP,NV,NDATA
1183        WRITE (66,FMT=190,IOSTAT=N,ERR=1070) (PARAM(1,I),I=1,NP)
1184        WRITE (66,FMT=*,IOSTAT=N,ERR=1070) (VARIABLE(1,I),I=1,NV)
1185        IF (Code.EQ.'D') THEN
1186          WRITE (66,FMT=*,IOSTAT=N,ERR=1070) (Z(I),I=1,NDATA)
1187          GO TO 1060
1188        END IF
1189        WRITE (66,FMT=*,IOSTAT=N,ERR=1070) (Y(I),I=1,NDATA)
1190   1060 WRITE (66,FMT=*,IOSTAT=N,ERR=1070) (X(I),I=1,NDATA)
1191        CLOSE (UNIT=66,IOSTAT=N,ERR=1070,STATUS='KEEP')
1192        IF (Code.EQ.'D') Dstat='STORED'
1193        IF (Code.EQ.'S') Fstat='STORED'
1194        CALL Wait(2.0)
1195        GO TO MENU
1196 C      ------------------------------------------------------------
1197   1070 WRITE (1,*) UP,ERASE,'  Disk Error #',N,BELL,' :_'
1198        WRITE (1,*) 'R...Resave, X...EXIT to Menu ? ',BELL,'_'
1199        READ (1,80) Icode
1200        CALL Upper(Icode)
1201        IF ((Icode.NE.'R').AND.(Icode.NE.'X')) GO TO 1070
1202        WRITE (1,*) UP,ERASE,'  Deleting Old File: ',Filespec(1:K),BELL,
1203       &' _'
1204        CLOSE (UNIT=66,IOSTAT=N,ERR=1070,STATUS='DELETE')
1205        CALL Wait(2.0)
1206        IF (Icode.EQ.'R') GO TO 1020
1207        Fname=' '
1208        Fstat=' '
1209        GO TO MENU
```

```
C
C     ---------------------------------------------------------------
C
C         Store Curve Fitting File
C
C     ---------------------------------------------------------------
C
 2000 WRITE (1,*) HOME,CLR,'_'
      TITLE='Curve Fitting File'
      CALL Center(TITLE)
      CALL Line(NCOL)
      WRITE (1,*) DOWN,'  Scan Range: ',WMAX,' to',WMIN,' nm',DOWN,
     &DOWN
 2010 WRITE (1,*) UP,ERASE,'  File Range: (MIN,MAX) ? ',BELL,'_'
      READ (1,*,ERR=2010) WN,WX
      WN=ABS(WN)
      WX=ABS(WX)
      CALL Xlimits(WN,WX)
      IF ((WN.LT.WMIN).OR.(WN.GT.WMAX)) GO TO 2010
      IF ((WX.LT.WMIN).OR.(WX.GT.WMAX)) GO TO 2010
C     ---------------------------------------------------------------
      T7=1.0E7
      WRITE (1,*) DOWN
 2020 WRITE (1,*) UP,ERASE,'  # Of Data Points: (10-500) ? ',BELL,'_'
      READ (1,'(I3)',ERR=2020) NFIT
      IF ((NFIT.LT.10).OR.(NFIT.GT.500)) GO TO 2020
      IF (NFIT.GT.NDATA) THEN
        WRITE (1,*) UP,ERASE,'  File Has Insufficient # Of Points: ',
     &  BELL,'_'
        CALL Wait(2.0)
        GO TO 2020
      END IF
      FIRST=T7/WX
      LAST=T7/WN
      SPACE=ANINT((LAST-FIRST)/FLOAT(NFIT-1))
 2030 LAST=FIRST+FLOAT(NFIT-1)*SPACE
      IF (T7/LAST.LT.WMIN) THEN
        SPACE=SPACE-1.0
        GO TO 2030
      END IF
      WRITE (1,*) UP,ERASE,UP,UP,ERASE,'_'
      WRITE (1,*) ' File Range: ',FIRST,' to',LAST,' at',SPACE,
     &' cm-1 steps',DOWN,DOWN
      EMULT=1.0
 2040 WRITE (1,*) UP,ERASE,'  Y-scale:  (A...Absorbance, ',
     &'E...Ext.Coeff.) ? ',BELL,'_'
      READ (1,80) Icode
      CALL Upper(Icode)
      IF (Icode.EQ.'E') THEN
        EMULT=1.0/(CONC(1)*PATH(1))
        WRITE (1,*) UP,ERASE,'  Ext.Coeff. vs cm-1'
        GO TO 2050
      END IF
```

```
1263        IF (Icode.NE.'A') GO TO 2040
1264        WRITE (1,*) UP,ERASE,'  Absorbance vs cm-1'
1265 C      ------------------------------------------------------------
1266   2050 WRITE (1,*) DOWN
1267        IF (Ymode.EQ.'T') THEN
1268          WRITE (1,*) UP,ERASE,'  %T Data -> Absorbance'
1269          WRITE (1,*) DOWN
1270        END IF
1271   2060 WRITE (1,*) UP,ERASE,'  Filename: ? ',BELL,'_'
1272        READ (1,'(A20)') Fname
1273        OPEN (UNIT=66,FILE=Fname,IOSTAT=N,STATUS='NEW')
1274        IF (N.NE.0) THEN
1275          N=N-500
1276          WRITE (1,*) UP,ERASE,'_'
1277          IF (N.EQ.2) WRITE (1,*) ' FILE EXISTS: ',BELL
1278          IF (N.EQ.8) WRITE (1,*) ' FILE OPENED: ',BELL
1279          CLOSE (66,IOSTAT=N,ERR=2999)
1280          CALL Wait(2.0)
1281          GO TO 2060
1282        END IF
1283        WRITE (1,*) UP,ERASE,'  Writing File: ',Fname,BELL
1284 C      ------------------------------------------------------------
1285        DO 2070 W=FIRST,LAST,SPACE
1286          WL=ANINT(T7/W/STEP)*STEP       ! Nearest Wavelength
1287          I=NINT((WMAX-WL)/STEP)+1       ! Index Value
1288          Xval=T7/X(I)                   ! Nearest Wavenumber
1289          IF (Ymode.EQ.'A') THEN
1290            Yval=Y(I)*EMULT              ! Absorbance/Ext.Coeff. Value
1291          ELSE
1292            Yval=ALOG10(100.0/Y(I))*EMULT
1293          END IF
1294          WRITE (66,*) Xval,Yval
1295   2070 CONTINUE
1296 C      ------------------------------------------------------------
1297        CLOSE (66,IOSTAT=N,ERR=2999,STATUS='KEEP')
1298        Cstat='STORED'
1299        GO TO MENU
1300 C      ------------------------------------------------------------
1301   2999 WRITE (1,*) '  Disk Error #',N
1302        CALL Wait(2.0)
1303        Cstat='INVALID'
1304        GO TO MENU
1305 C
1306 C      ------------------------------------------------------------
1307 C
1308 C          Exit Program
1309 C
1310 C      ------------------------------------------------------------
1311 C
1312   9999 IF ((Sstat.EQ.'VALID').AND.(Fstat.NE.'STORED')) THEN
1313          WRITE (1,*) UP,ERASE,'_'
1314          WRITE (1,*) ' Spectrum NOT STORED: Exit, (Y or N) ? ',BELL,'_'
1315          READ (1,80) Icode
```

```
1316          CALL Upper(Icode)
1317          IF (Icode.EQ.'N') GO TO 70
1318          IF (Icode.NE.'Y') GO TO 9999
1319        END IF
1320        WRITE (1,*) UP,ERASE,UP
1321        STOP
1322        END
1323 C
1324 C     ******************** END OF MAIN PROGRAM ************************
1325 C
1326 C         Print a TITLE Centered in 72 columns
1327 C
1328 C     ***********************************************************************
1329 C
1330        SUBROUTINE Center(TITLE)
1331        INTEGER I,J,N
1332        CHARACTER TITLE*72,BLANK*36
1333        BLANK='                                    '
1334        I=72
1335        J=0
1336        DO WHILE (TITLE(I:I).EQ.' ')
1337          J=J+1
1338          I=72-J
1339        END DO
1340        N=J/2
1341        WRITE (1,*) BLANK(1:N),TITLE(1:I)
1342        RETURN
1343        END
1344 C
1345 C     -----------------------------------------------------------------
1346 C
1347 C         Print a line of N '-' characters (72 columns max)
1348 C
1349 C     -----------------------------------------------------------------
1350 C
1351        SUBROUTINE Line(N)
1352        INTEGER I,N
1353        CHARACTER BLANK*72,DLINE*72,SPACE*36
1354        SPACE='                                    '
1355        BLANK=SPACE//SPACE
1356        SPACE='------------------------------------'
1357        DLINE=SPACE//SPACE
1358        IF (N.GT.72) N=72
1359        I=(72-N)/2
1360        WRITE (1,*) BLANK(1:I),DLINE(1:N)
1361        RETURN
1362        END
```

```
C
C     ------------------------------------------------------------
C
C         Validate Abscissa Limits
C
C     ------------------------------------------------------------
C
      SUBROUTINE Xlimits(MIN,MAX)
      REAL MIN,MAX,SWAP
      MIN=ABS(MIN)
      MAX=ABS(MAX)
      IF (MIN.LT.MAX) GO TO 10
      SWAP=MIN
      MIN=MAX
      MAX=SWAP
   10 RETURN
      END
C
C     ------------------------------------------------------------
C
C         Validate Ordinate Limits
C
C     ------------------------------------------------------------
C
      SUBROUTINE Ylimits(MIN,MAX)
      REAL MIN,MAX,SWAP
      IF (MIN.LT.MAX) GO TO 10
      SWAP=MIN
      MIN=MAX
      MAX=SWAP
   10 RETURN
      END
C
C     ------------------------------------------------------------
C
C         Convert String Entry To Upper Case If Required
C
C     ------------------------------------------------------------
C
      SUBROUTINE Upper(Code)
      INTEGER LENSTR,N
      CHARACTER*(*) Code
      LENSTR=LEN(Code)
      DO 10 I=1,LENSTR
        N=ICHAR(Code(I:I))
        IF (N.GT.96) Code(I:I)=CHAR(N-32)
   10 CONTINUE
      RETURN
      END
```

```
C
C     ---------------------------------------------------------------
C
C         Convert ASCII String To Numeric Value (10 Digits Max'm)
C
C     ---------------------------------------------------------------
C
      SUBROUTINE Val(String,VALUE)
      INTEGER DECPT,EXPON,LENSTR,M,N,NUM(12),EX(3)
      REAL VALUE
      DOUBLE PRECISION ESIGN,MULT,SIGN,TEN,DECIMAL
      CHARACTER Ascii
      CHARACTER*(*) String
      LOGICAL EXPONENT,INTEGER,TEST
      EXPONENT=.FALSE.
      INTEGER=.TRUE.
C     TEST=.FALSE.
      J=1
      K=0
      M=0
      DECPT=0
      ESIGN=1.0
      SIGN=1.0
      TEN=10.0
      DECIMAL=0.0
      LENSTR=LEN(String)
C     IF (TEST) WRITE (1,*) ' String Number = ',String
C     IF (TEST) WRITE (1,*) ' String Length =',LENSTR
      DO 100 I=1,LENSTR
        Ascii=String(I:I)
        N=ICHAR(Ascii)
        IF (N.EQ.69) THEN
          EXPONENT=.TRUE.
          IPOS=I
          GO TO 100
        END IF
        IF (EXPONENT) GO TO 30
        IF ((N.GE.48).AND.(N.LE.57)) GO TO 20
        IF (N.EQ.46) INTEGER=.FALSE.
        IF (N.EQ.46) DECPT=K
        IF (N.EQ.45) SIGN=-1.0
        GO TO 100
   20 NUM(J)=N-48
      K=J
      J=J+1
      GO TO 100
   30 IF (N.EQ.45) ESIGN=-1.0
      IF ((N.LT.48).OR.(N.GT.57)) GO TO 100
      EX(I-IPOS)=N-48
      M=M+1
  100 CONTINUE
      IF ((DECPT.EQ.0).AND.(INTEGER)) DECPT=K
```

```
1464        DO 200 J=1,K
1465          EXPON=DECPT-J
1466          MULT=TEN**EXPON
1467          DECIMAL=DECIMAL+NUM(J)*MULT
1468    200 CONTINUE
1469        DECIMAL=SIGN*DECIMAL
1470        MULT=0.0
1471        IF (ESIGN.EQ.-1.0) M=M+1
1472        DO 300 I=1,M
1473          MULT=MULT+DBLE(EX(I))*TEN**(M-I)
1474    300 CONTINUE
1475        VALUE=DECIMAL*TEN**(ESIGN*MULT)
1476 C      IF (TEST) WRITE (1,*) ' Value =',VALUE
1477        RETURN
1478        END
```

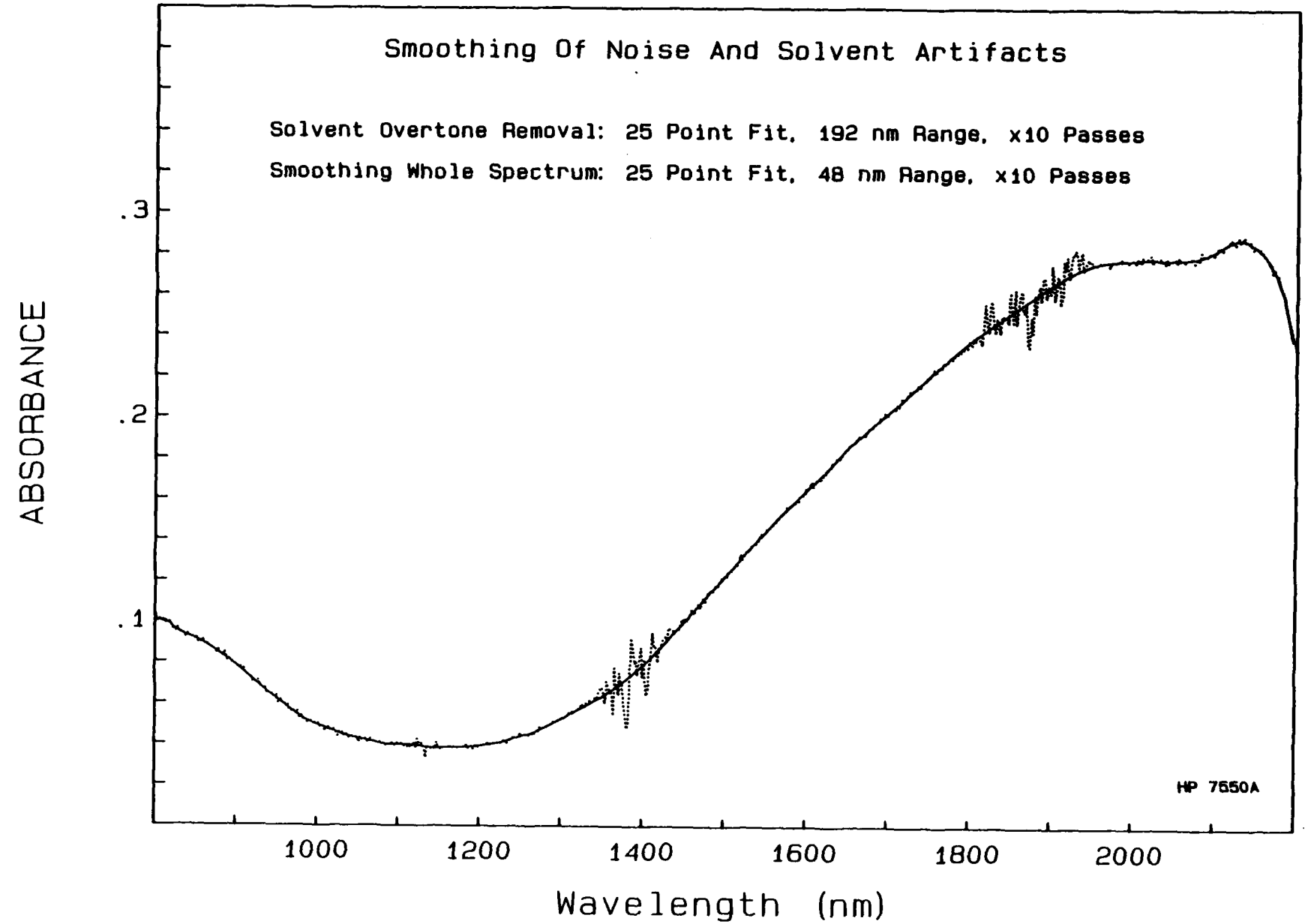
```
1479 C
1480 C     ---------------------------------------------------------------
1481 C
1482 C         Convert Number To ASCII String
1483 C
1484 C     ---------------------------------------------------------------
1485 C
1486       SUBROUTINE Str(VALUE,String,PREC)
1487       INTEGER ASCII,DECPT,I,J,LENSTR,NDIGIT,NUMBER,PREC
1488       REAL VALUE
1489       DOUBLE PRECISION DECIMAL,FRACTION,TEN
1490       CHARACTER Concat*14,Digit(12),Sign,String*14
1491       LOGICAL INTEGER,TEST
1492       INTEGER=.TRUE.
1493 C     TEST=.FALSE.
1494       DECPT=0
1495       J=0
1496       TEN=10.0
1497       Sign=' '
1498       Concat=' '
1499 C     IF (TEST) WRITE (1,*) ' Value Entered = ',VALUE
1500       IF (VALUE.LT.0.0) Sign='-'
1501       IF (VALUE.EQ.0.0) GO TO 100
1502       DECIMAL=ABS(VALUE)
1503       DO WHILE (DECIMAL.GE.1.0)
1504         DECIMAL=DECIMAL/TEN
1505         J=J+1
1506       END DO
1507       DECPT=J
1508 C     IF (TEST) WRITE (1,*) ' # of Whole Digits: ',DECPT
1509       IF (DECPT.EQ.0) GO TO 30
1510       DO 20 J=1,DECPT
1511         DECIMAL=DECIMAL*TEN
1512         NUMBER=INT(DECIMAL)
1513         ASCII=NUMBER+48
1514         Digit(J)=CHAR(ASCII)
1515         FRACTION=DECIMAL-NUMBER
1516         DECIMAL=DINT(FRACTION*TEN**(PREC-J)+.5)/TEN**(PREC-J)
1517    20 CONTINUE
1518 C     IF (.NOT.TEST) GO TO 30
1519 C     WRITE (1,*) ' The Whole Digits = ',(Digit(I), I=1,DECPT)
1520    30 J=DECPT
1521 C     IF (TEST) WRITE (1,*) ' Decimal Fraction = ',DECIMAL
1522       IF (DECIMAL.NE.0.0) INTEGER=.FALSE.
1523       IF (DECPT.GE.12) GO TO 40
1524       DO WHILE (DECIMAL.NE.0.0)
1525         J=J+1
1526         DECIMAL=DECIMAL*TEN
1527         NUMBER=INT(DECIMAL)
1528         ASCII=NUMBER+48
1529         Digit(J)=CHAR(ASCII)
1530         FRACTION=DECIMAL-NUMBER
1531         DECIMAL=DINT(FRACTION*TEN**(PREC-J)+.5)/TEN**(PREC-J)
```

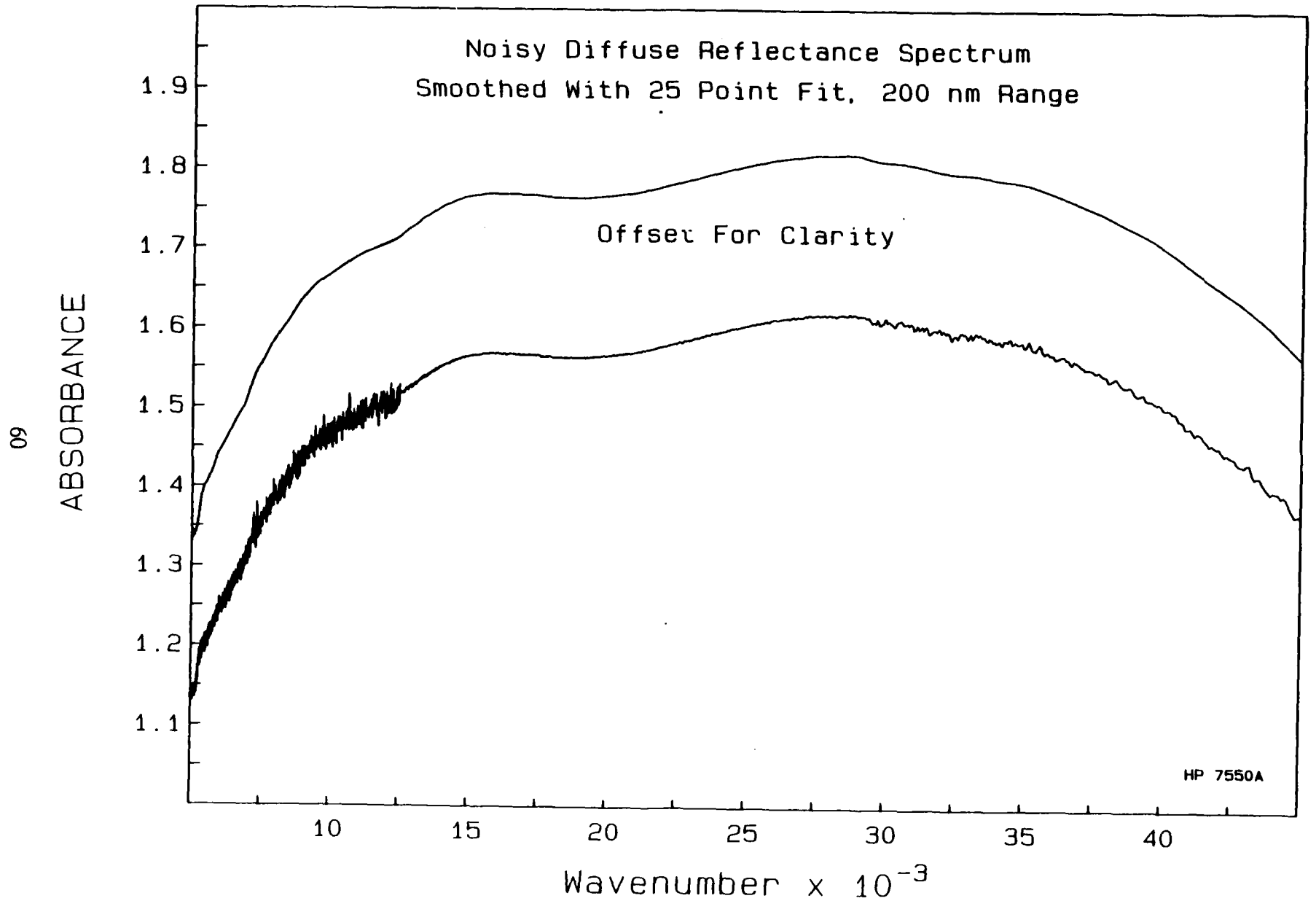
```
1532          IF (DECIMAL.EQ.1.0) THEN
1533            DIGIT(J)=CHAR(ASCII+1)
1534            DECIMAL=0.0
1535          END IF
1536          IF (J.GE.12) DECIMAL=0.0
1537        END DO
1538     40 NDIGIT=J
1539 C      IF (.NOT.TEST) GO TO 50
1540 C      WRITE (1,*) ' The Characters  = ',(Digit(I), I=1,NDIGIT)
1541     50 IF (NDIGIT.GT.12) GO TO 200
1542        DO 60 I=1,NDIGIT
1543          Concat(I:I)=Digit(I)
1544     60 CONTINUE
1545        IF (INTEGER) GO TO 80
1546        IF (DECPT.EQ.0) GO TO 70
1547        String=Sign//Concat(1:DECPT)//'.'//Concat(DECPT+1:14)
1548        RETURN
1549     70 String=Sign//'.'//Concat
1550        RETURN
1551     80 String=Sign//Concat
1552        RETURN
1553    100 String=' 0.0'
1554        RETURN
1555    200 WRITE (1,*) ' Error in data: (too many digits)'
1556        STOP
1557        END
```

```
1558 C
1559 C     ------------------------------------------------------------------
1560 C
1561 C         Wait Specified Delay (sec)
1562 C
1563 C     ------------------------------------------------------------------
1564 C
1565       SUBROUTINE Wait(DELAY)
1566       REAL DELAY,PERIOD,Tzero,Time
1567       PERIOD=0.0
1568       Tzero=Time(I)
1569       DO WHILE (PERIOD.LT.DELAY)
1570         PERIOD=Time(I)-Tzero
1571       END DO
1572       RETURN
1573       END
1574 C
1575 C     ------------------------------------------------------------------
1576 C
1577 C         Read Time (sec) from the HP 1000's RTE-6 Operating System
1578 C
1579 C         Note: I is a dummy argument, no values are passed
1580 C
1581 C     ------------------------------------------------------------------
1582 C
1583       REAL FUNCTION Time(I)
1584       INTEGER ICODE,ITIME(5)
1585       ICODE=11
1586       CALL EXEC(ICODE,ITIME)
1587       Time=FLOAT(ITIME(1))/100.0+FLOAT(ITIME(2))+FLOAT(ITIME(3))*60.0
1588      &+FLOAT(ITIME(4))*3600.0
1589       RETURN
1590       END
```

Appendix I

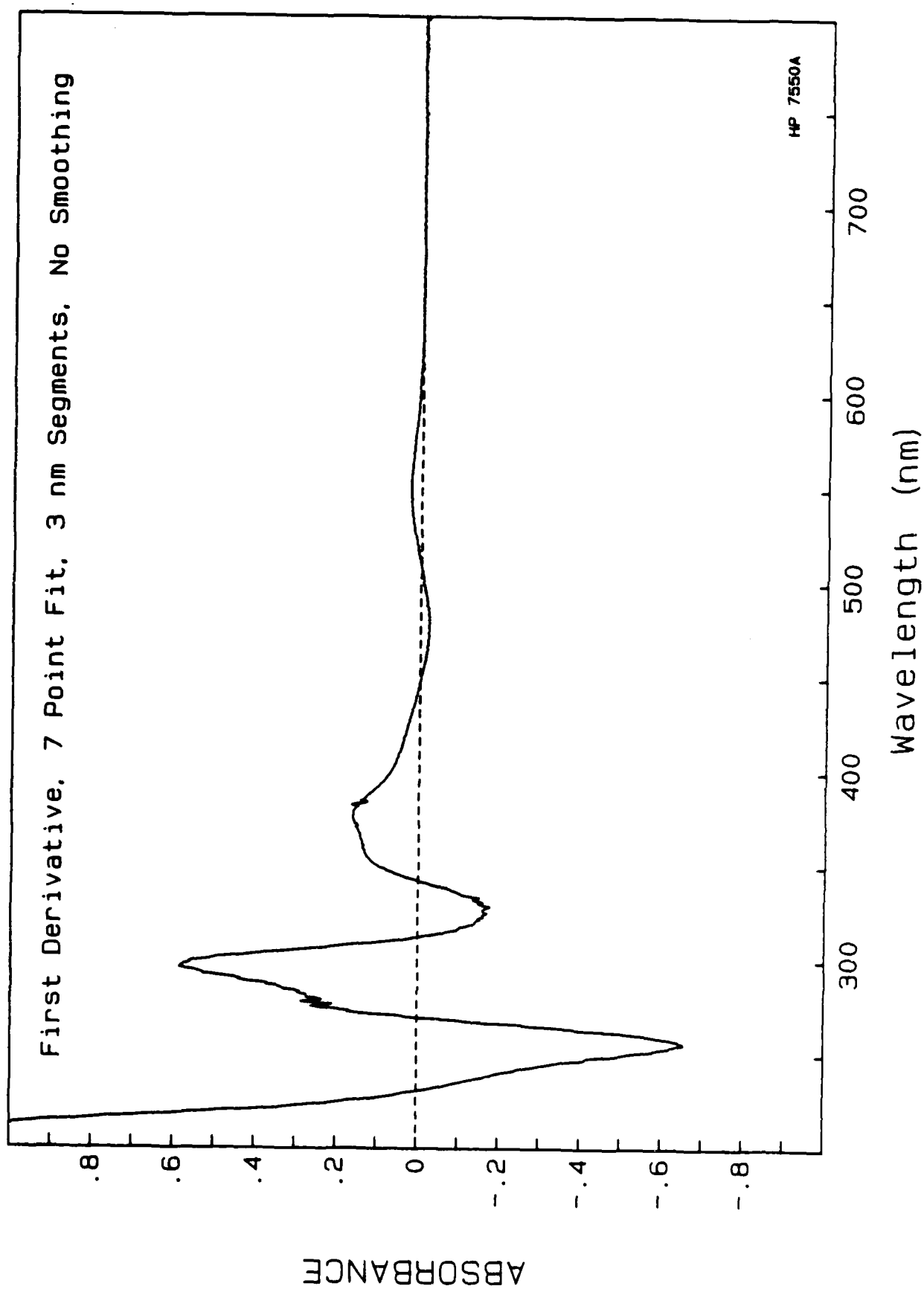
Sample Smoothed Spectra

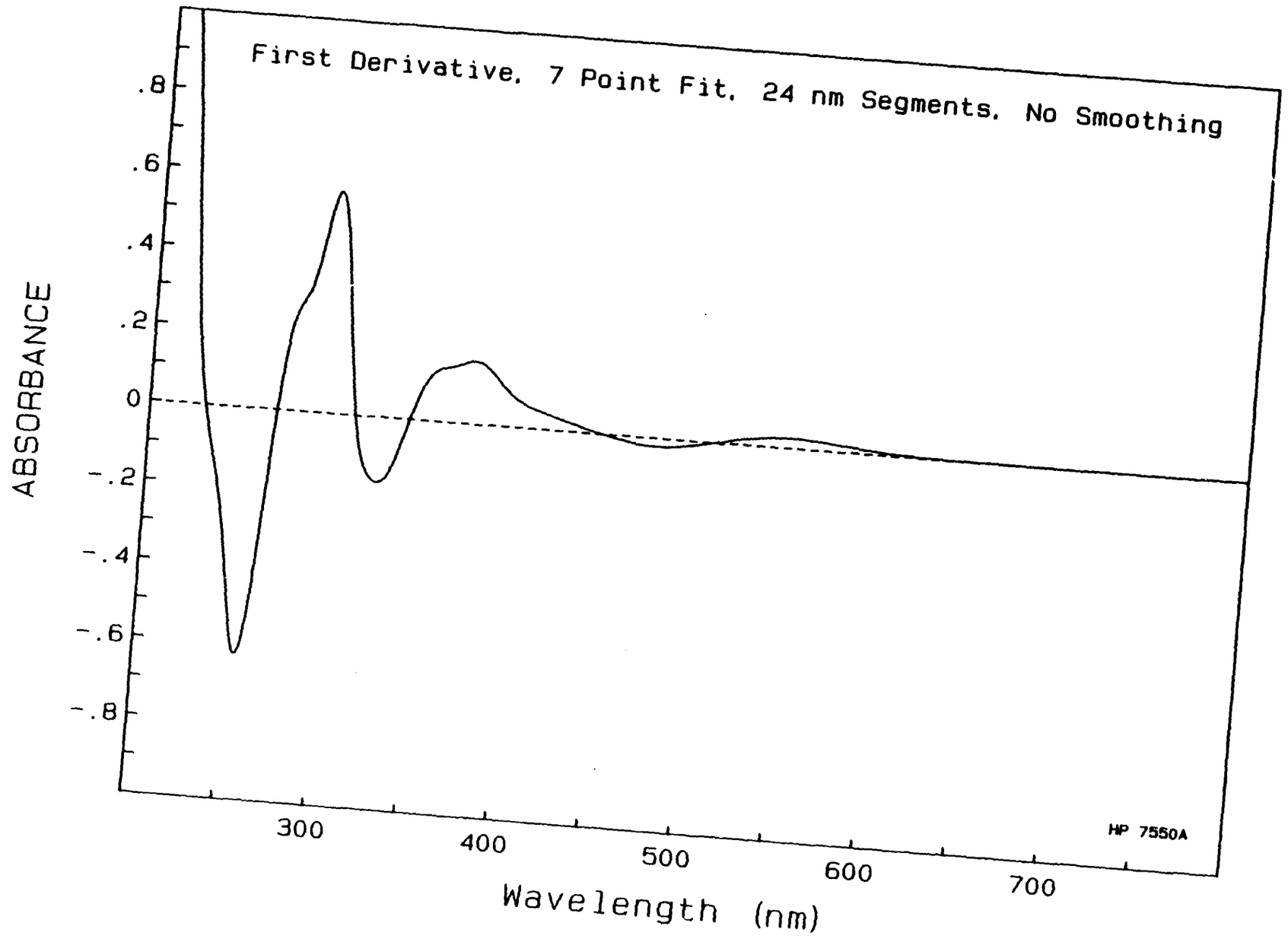

Smoothing Of Noise And Solvent Artifacts
Solvent Overtone Removal: 25 Point Fit, 192 nm Range, x10 Passes
Smoothing Whole Spectrum: 25 Point Fit, 48 nm Range, x10 Passes
ABSORBANCE
.3
.2
.1
1000
1200
1400
1600
1800
2000
Wavelength (nm)
HP 7550A

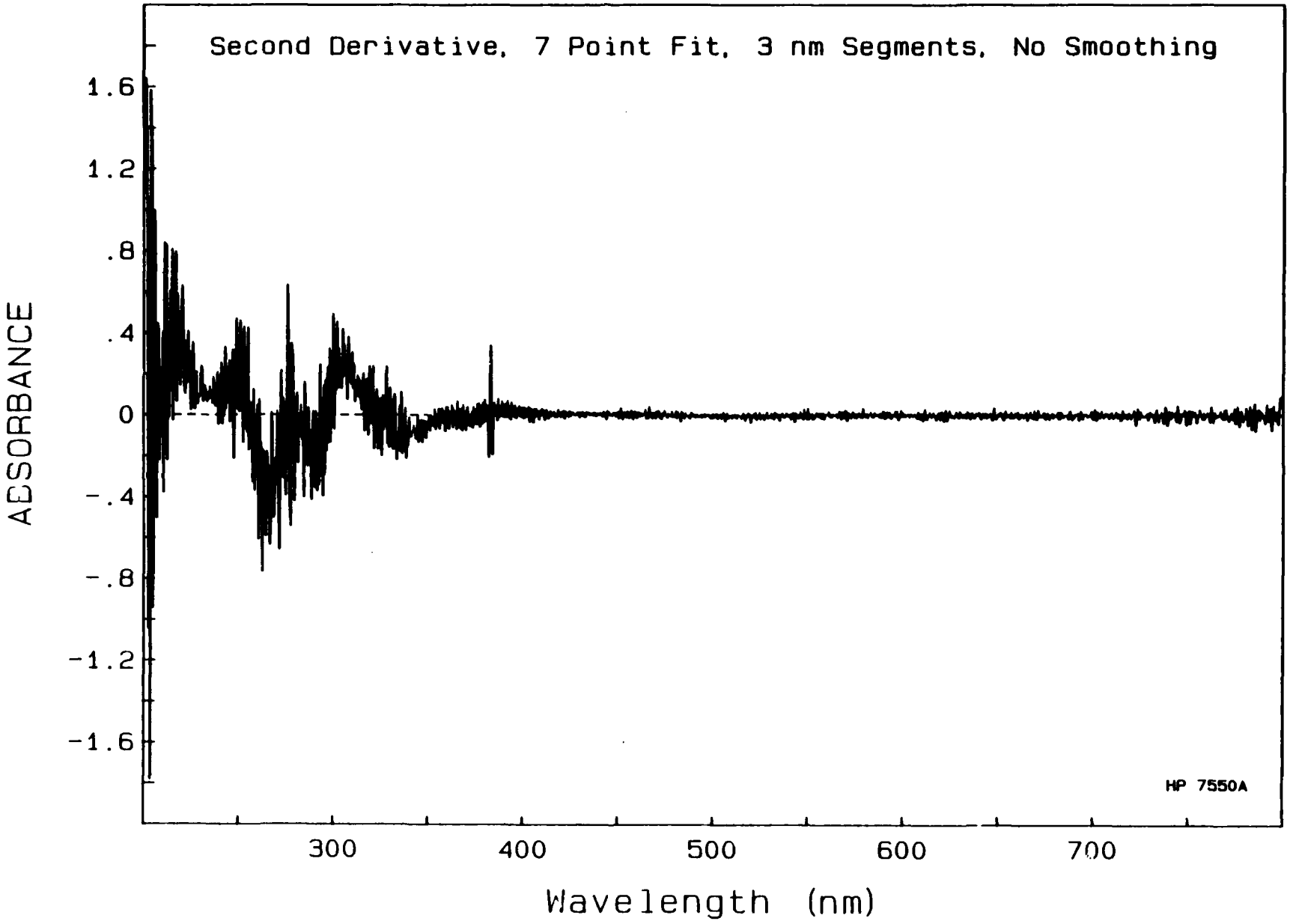

Noisy Diffuse Reflectance Spectrum
Smoothed With 25 Point Fit, 200 nm Range
Offset For Clarity
ABSORBANCE
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
HP 7550A
10
15
20
25
30
35
40
Wavenumber x 10^{-3}

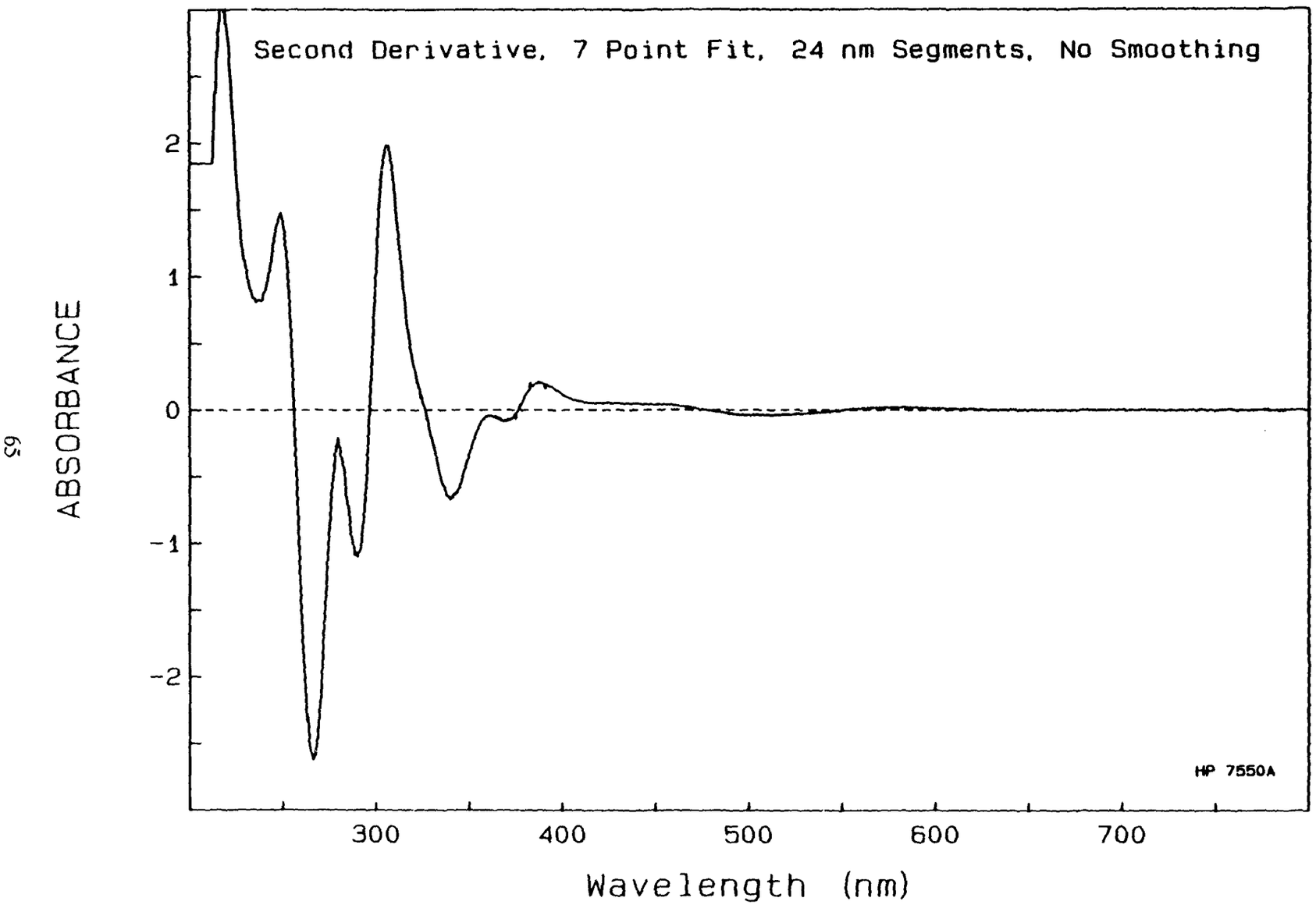
Appendix II

Sample Derivative Spectra


First Derivative, 7 Point Fit, 3 nm Segments, No Smoothing
HP 7550A
ABSORBANCE
.8
.6
.4
.2
0
-.2
-.4
-.6
-.8
300
400
500
600
700
Wavelength (nm)


First Derivative, 7 Point Fit, 24 nm Segments, No Smoothing
ABSORBANCE
.8
.6
.4
.2
0
-.2
-.4
-.6
-.8
300
400
500
600
700
Wavelength (nm)
HP 7550A


Second Derivative, 7 Point Fit, 3 nm Segments, No Smoothing
ABSORBANCE
1.6
1.2
.8
.4
0
-.4
-.8
-1.2
-1.6
300
400
500
600
700
Wavelength (nm)
HP 7550A


Second Derivative, 7 Point Fit, 24 nm Segments, No Smoothing
ABSORBANCE
2
1
0
-1
-2
300
400
500
600
700
Wavelength (nm)
HP 7550A